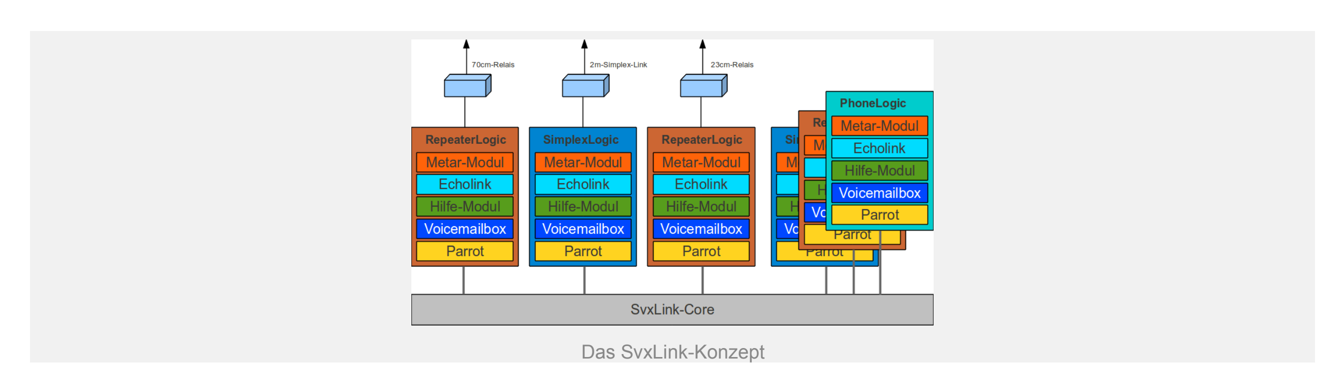
Allgemeines zu SVX-Link

Konzept

SvxLink wurde von Tobias (SM0SVX) bereits um 2002 ins Leben gerufen und wird seitdem ständig weiterentwickelt.
Es besteht aus einem Haupt- bzw. Kernprogramm und sogenannten Logiken, die quasi beliebig kombiniert werden können. Dabei kann jeder Logik verschiedene Module zugeordnet werden. Echolink ist dabei nur eines von weiteren Modules, somit unterscheidet sich SvxLink vom klassischen Echolink ganz erheblich.
Man ist dabei nicht auf den Betrieb eines einzelnen Relais/Simplex-Link beschränkt, sondern kann theoretisch beliebig viele Relais/Links unabhängig voneinander konfigurieren und betreiben. Das folgende Bild soll das Prinzip etwas verdeutlichen.



Das SvxLink-Konzept

Wenn es die Soundkarte unterstützt, dann lassen sich die beiden Stereokanäle getrennt nutzen, d.h. der linke Kanal steuert das 70cm-Relais, der rechte Kanal das 23cm-Relais und beide selbstverständlich unabhängig voneinander. Grenzen sind hier nur (irgendwann einmal) durch die Rechnerperformance gesetzt.
Prinzipiell gibt es bei dieser Applikation zwei unterschiedliche Ebenen, einmal das Kernsystem, welches die grundlegenden Funktionen realisiert und eine Art Präsentationsebene, die die Funktionalität nach “außen” hin bereitstellt.
Die Präsentationsebene wird durch ein in SvxLink implementiertes Ereignisbehandlungssystem realisiert. Bei jedem nur möglichen Ereignis wie z.B. Öffnen oder Schließen der Rauschsperre, eingehende Echolinkverbindung, … wird eine TCL-Funktion aufgerufen, in die der SvxLink-Sysop eigene Routinen einbauen kann. Diese Schnittstelle ist grundsätzlich vergleichbar mit der VisualBasic-Erweiterung bei EcholinkPlus, bietet aber ungleich mehr Funktionalität.

SvxLink-Features

Auf diese Seite finden sie eine Übersicht über die Funktionen des SvxLink-Servers sowie Qtel. Sie dient der schnellen Übersicht, wofür die Programme genutzt werden können.

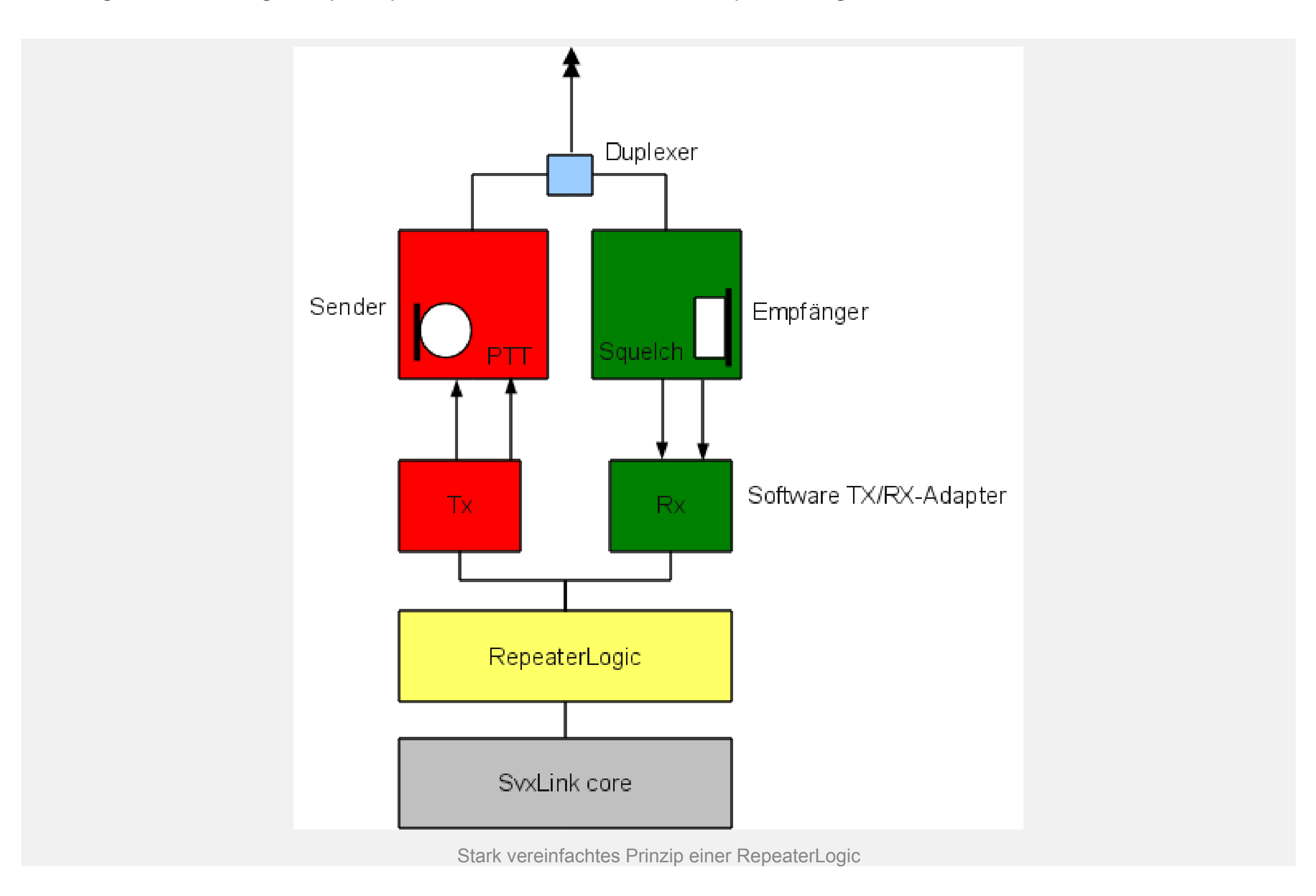
Der SvxLink-Server
Das sind die wichtigsten Funktionen des SvxLink-Servers:

Kann als Relais-Controller fungieren oder als einfacher Simplex-Link
Modulbasiertes System, um Sprachservices in den Programmkern nachladen zu können
steuerbar per DTMF-Töne
DTMF – Anti-Flatter-Modus, um bei Mobilbetrieb und/oder schwachen Signalstärken einen DTMF-Ton nicht mehrfach zu erkennen
Sound-Dateien für Ansagen und Informationen
Periodische Identifizierung frei programmierbar

Ein TCL-basiertes System, das bestimmte Ereignisse speziell behandeln kann
Stereo-Kanäle können wie einzelne Mono-Kanäle behandelt werden
Roger-Piep
Zusammenschaltung von verschiedenen Logiken (Repeater, Links) per Kommando
Multi-Tone-Encoder nach ZVEI-xx, CCITT, EIA, Natel, Motorola, …
Multi-Tone-Decoder als Macro nutzbar
Manuelle Identifizierung beim DTMF-Ton *
einfaches Makrosystem für Kurzwahlfunktionen
APRS-Status- und Baken-Funktionen
RemoteTRX erlaubt Vernetzung von verschiedenen RX und TX
Unterstützung von EchoLink-Proxyservern

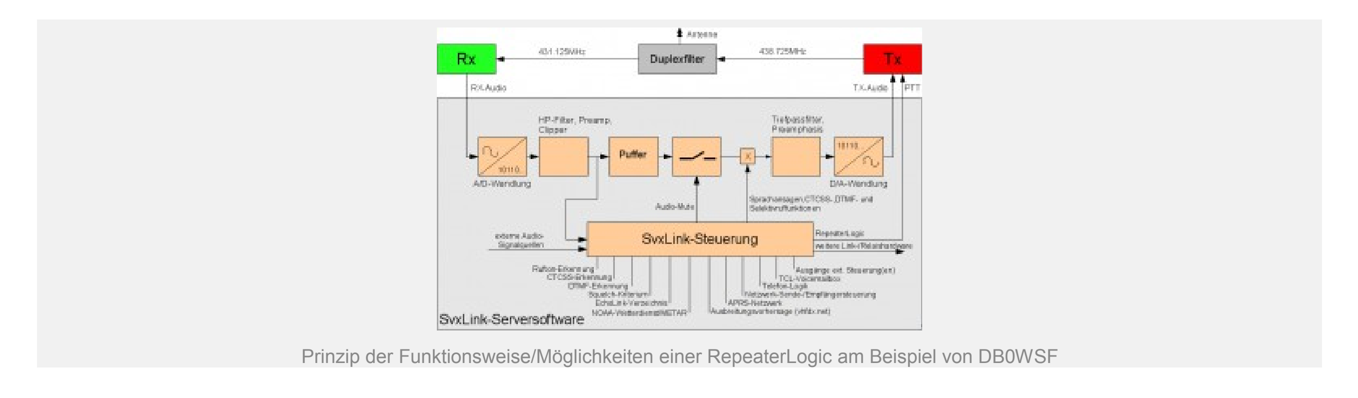
RepeaterLogic

Das folgende Bild zeigt die prinzipielle Funktionsweise einer RepeaterLogic.



Stark vereinfachtes Prinzip einer RepeaterLogic

Im folgenden Bild sind die Aufgaben etwas dataillierter dargestellt. Nicht alle Funktionen sind für eine RepeaterLogic notwendig. Es soll nur den möglichen Funktionsumfang von SvxLink verdeutlichen.



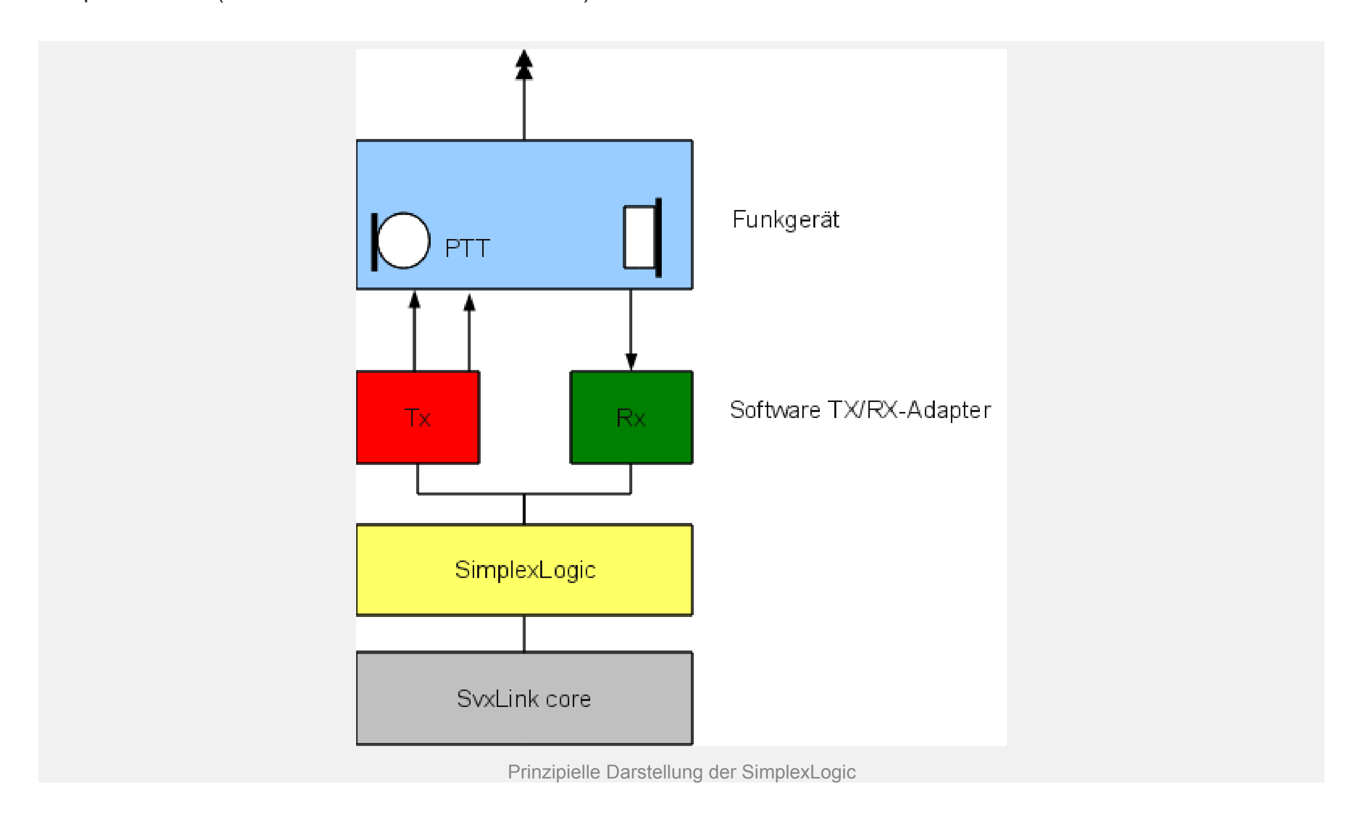
Prinzip der Funktionsweise/Möglichkeiten einer RepeaterLogic am Beispiel von DB0WSF

Funktionen der RepeaterLogic

verschiedene Aktivierungsmöglichkeiten wie 1750Hz-Ton, CTCSS, DTMF oder per Rauschsperre
Periodischer “idle-tone” wenn das Relais aufgetastet ist aber kein Signal empfangen wird
Ansagen werden mit der normalen Sprachübertragung gemischt
Die Lautstärke der Ansagen wird bei anderen Sprachübertragungen vermindert
Möglichkeit, mehrere Kernprogramme zu definieren um mehrere Funkgeräte anzuschließen
Kernprogramme können miteinander verbunden werden
Unterstützung für mehrere Empfänger, die über einen eingebaute Software automatisch zu- oder abgeschaltet werden können (Auswahl des RX mit bester Signalqualität)
Netzwerk-Empfänger können per TCP/IP mit dem Kernprogramm verbunden werden
Netzwerk-Sender können per TCP/IP mit dem Kernprogramm verbunden werden
Mehrere Squelsh-Detektoren: VOX, CTCSS, Signalerkennung über die serielle Schnittstelle
Höhenanhebung/-absenkung auf dem Übertragungsweg
DTMF-Unterdrückung
Rauschenunterdrückung vor dem Einsetzen der Rauschsperre
Senden von CTCSS-Subaudiotönen, ständig oder abhängig vom Status der Rauschsperre
einfaches Hilfemodul in englischer/deutscher Sprache
Parrot-Modul, das das empfangene wieder aussendet (“Papagei”)
Echolink-Modul das die Teilnahme am weltweiten Echolink-Netz ermöglicht
Leerlauf-Zeitbegrenzung
Rufzeichensuche bzw. Verbindung durch Rufzeicheneingabe
Auflistung verbundener Stationen durch Kommando
Funktion zur Ansage der eigenen Node-ID
Verbindungsaufnahme zu Relais, Konferenzen oder Links per Zufall möglich
Wiederverbinden mit der letzten Station
SWL-Modus (nur Zuhören)
Öffnen des Repeaters durch Squelch, Rufton, CTCSS, Mehrton(ZVEI,EIA,…) und/oder DTMF

SimplexLogic

Eine SvxLink-SimplexLogic entspricht einem Link (DL1ABC-L) bei Echolink. Kennzeichnend ist der reine Simplexbetrieb (wie der Name schon beschreibt).



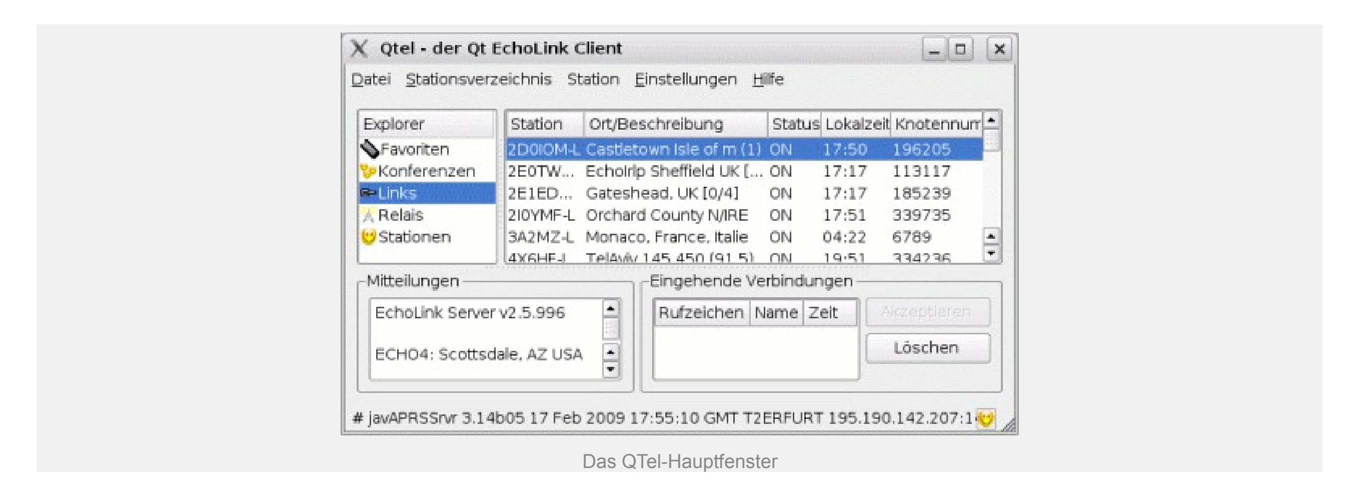
Prinzipielle Darstellung der SimplexLogic

Ein einzelnes Funkgerät wird an die Soundkarte (mit Hilfe eines Adapters) angeschlossen. Die PTT-Ansteuerung liefert die TX-Logic mit. Schauen Sie unter *Details->Konfiguration->SimplexLogic* wie Sie solch einen einfachen SvxLink-Node konfigurieren können.

QTel-Client

Qtel ist die Abkürzung für “QT Echolink-Client”. Es handelt sich hierbei um einen reinen Echolink-Clienten, der keinen “Sysop-Modus” besitzt. Wenn Sie einen Link aufbauen möchten dann werfen Sie einen Blick auf den SvxLink-server. Qtel ist ein in der Bedienung sehr einfaches Programm und die meisten Dinge sind selbsterklärend. Inzwischen gibt es Sprachunterstützung für Englisch, Schwedisch, Deutsch, Italienisch, Holländisch und Spanisch. Übersetzungen für Türkisch und Ungarisch sind in Vorbereitung. Seit 5/2013 wird auch die Verwendung von EchoLink-Proxyservern unterstützt.

Es gibt drei Fenster: das Hauptfenster, der Konfigurationsdialog und der Kommunikationsdialog.



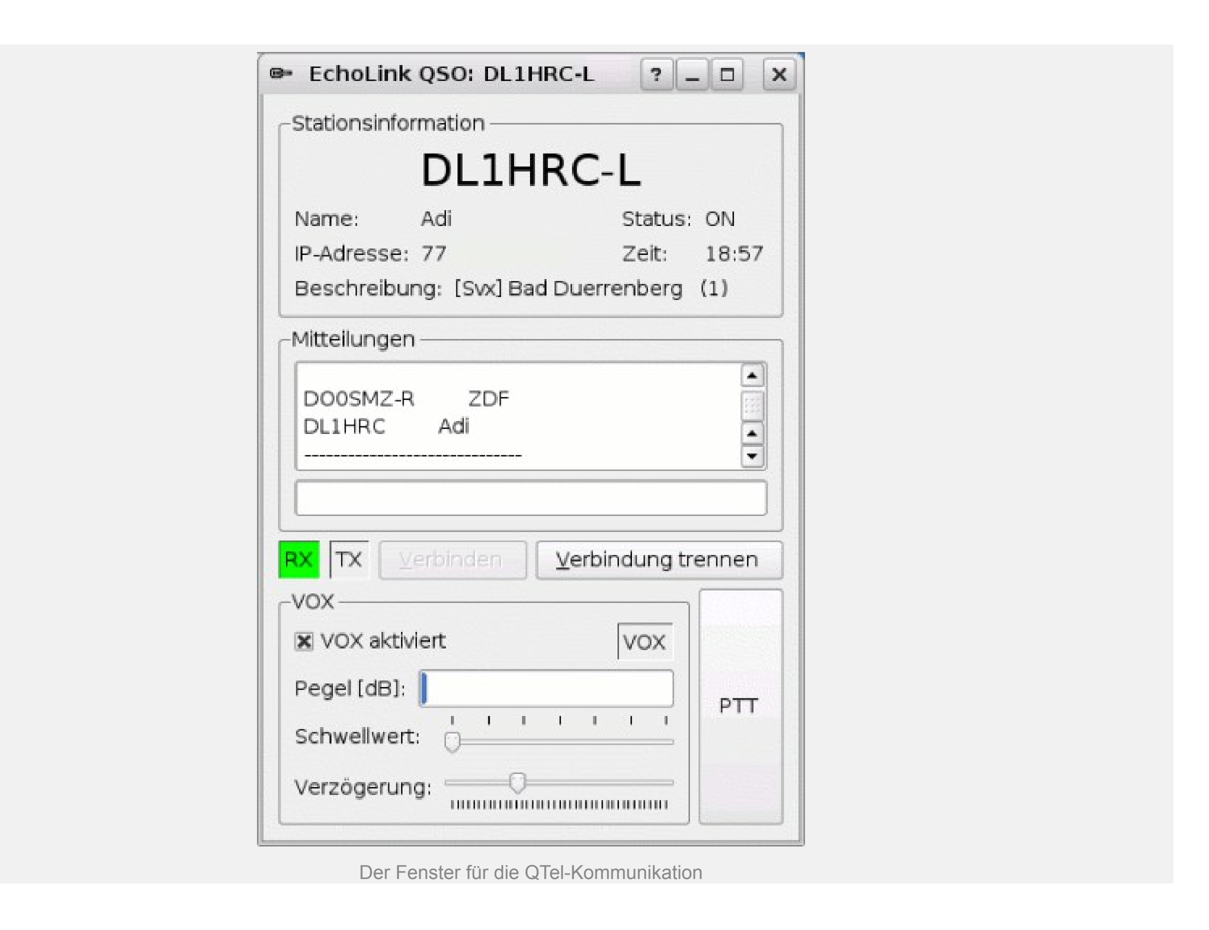
Das QTel-Hauptfenster

Das Hauptfenster besteht aus 4 Bereichen. Im oberen linken Bereich kann man auswählen, welche Stations-Kategorie gerade angezeigt werden soll. Alle Stationen einer Kategorie werden dann im rechten Fenster aufgelistet.
Es gibt 4 Kategorien: Konferenzen, Links, Relais und (private) Stationen. Die 5. Kategorie ist ein Bereich in dem alle Stationen selbsttätig eingetragen werden können, die möglicherweise interessant sind, analog zu den Begriffen Favoriten oder Bookmarks bei Webbrowsern. Klicken Sie einfach mit der rechten Maustaste auf eine für Sie interessante Station und fügen Sie diese per “Zu den Favoriten hinzufügen…” Ihrer Favoriten-Liste hinzu.

Im linken unteren Bereich befindet sich das Fenster für Mitteilungen, die vom Echolinkserver übermittelt werden. Im rechten unteren Bereich befindet sich eine Liste eingehender Verbindungen. Um eine eingehende Verbindung zu akzeptieren klicken Sie diese Verbindung einfach an und drücken Sie dann auf “Akzeptieren”.

Der Kommunikationsdialog ermöglicht die Herstellung einer Verbindung zu einer anderen Echolink-Station. Um diesen Dialog aufzurufen, klicken Sie doppelt auf eine Station oder drücken Sie einfach, wenn diese Station farblich hinterlegt dargestellt ist. Um eine Verbindung herzustellen klicken Sie auf die Schaltfläche “Verbinden…”. Wenn eine Audioübertragung von dieser Station erfolgt, so erscheint die RX-Anzeige hellgrün. Um zu Senden drücken Sie die Schaltfläche “PTT”. Dabei leuchtet die TX-Anzeige rot. Um dauerhaft auf Sendung zu gehen drücken Sie zuerst auf der Tastatur die “Ctrl”-Taste und bei gedrückter Taste die Schaltfläche “PTT”. Um die Verbindung zu trennen drücken Sie die Schaltfläche “Trennen…”.

Im großen weißen Bereich in der Mitte des Kommunikationsdialoges werden Mitteilungen der Gegenstation angezeigt. Auch die Chat-Mitteilungen der Gegenstation werden hier angezeigt. In dem schmalen Bereich unter dieser weißen Fläche können eigene Chat-Mitteilungen an die Gegenstation eingegeben werden.



Der Fenster für die QTel-Kommunikation

Der Einstellungsdialog ("Qtel – Einstellungen") wird beim ersten Programmstart automatisch aufgerufen damit die Eingabe der eigenen Daten vereinfacht bzw. ermöglicht wird. Er kann jederzeit über das Menü "Einstellungen->Qtel Einstellungen" wieder aufgerufen werden. Die meisten Feldbezeichnungen sind selbsterklärend. Ort (bzw. Location) ist die im Echolink-Verzeichnis angezeigte Zeichenkette. Die im Bereich Information/Mitteilung eingegebenen Daten werden an die jeweilige Gegenstelle beim Verbindungsaufbau übermittelt.
ACHTUNG: Wenn man den Eintrag bei Ort bzw. Location vergisst erfolgt keine Anmeldung beim Echolink-Verzeichnisserver mit der Fehlermeldung: "The directory server closed the connection before all data was received"



Das Konfigurationsfenster

In der Registerkarte "Verzeichnisserver" gibt es einige Einstellungsmöglichkeiten für die Verbindung zum Echolink-Verzeichnisdienst. Wenn Sie die Maus über diesen Bereich bewegen ohne diesen anzuklicken erhalten Sie eine kurze Hilfestellung.

Das ist eigentlich alles was über das Programm Qtel zu sagen ist. Es ist wirklich einfach.

Bedienungsanleitung

Inhalt

* Vorwort
* Einführung
* Das Modul Hilfe
* Das Modul Parrot (Sprachpapagei)
* Das Modul Echolink
o Connect by Callsign

* Das Modul Mailbox
o Neue Nachrichten lesen
o Neue Nachrichten versenden
o Nachsehen ob neue Nachrichten warten

* Das Modul Metar Information
o Die Suchfunktionen

* Das Modul Selektiv-Ruf (Encoder)
o Selektivruf Encoder Makro

* Selektiv-Ruf (Decoder)
* Das Modul Ausbreitungsmonitor
* Sprachumschaltung
o Momentan verfügbare Sprachen

Einführung

SvxLink ist ein Sprach-Service-System welches unter Linux läuft. Es ist für die Anwendung im Amateurfunk gedacht und verfügt über spezielle Funktionen. Zu diesen zählt unter anderem die Steuerung von Relaisstationen (Hardware in the loop) und der Betrieb eines Echolink Gateways.
Die Software verfügt über verschiedene Module welche zur Laufzeit geladen werden müssen um deren vollen Funktionsumfang nutzen zu können.
Deshalb soll im folgenden soll die Bedieung der Software durch den Funkamateur erläutert werden und auf die momentan vorhandnen Module der Software eingegangen werden.
Grundlegend ist zu sagen, dass jedes Kommando welches an den Link oder das Relais gesendet wird mit einer Raute (#) abgeschlossen werden muss. Weiterhin ist es möglich den Status der SvxLink Station durch das senden eines Sterns (*) abzufragen. Beim Senden des Status werden folgede Informationen gesendet:

Name der Station, Uhrzeit, evtl. zu sendender CTCSS Ton, aktives Modul und die Auforderung 0# zu senden um Hilfe zu erhalten.

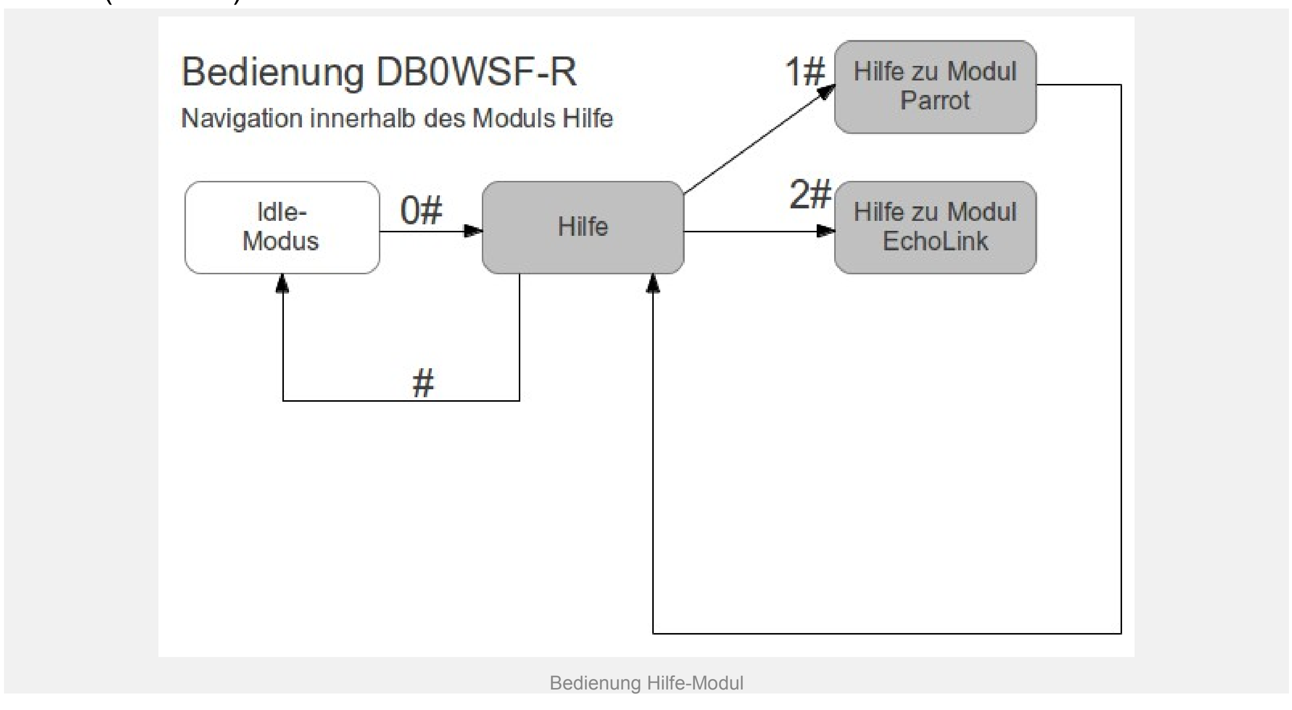
Diese hört sich z.B. wie folgt an:

“Delta Bravo Null Whisky Sierra Foxtrott Relais es ist jetzt Null Uhr sieben und fünfzig. Bitte einen CTCSS Ton von Einhundert Komma Null Hertz mitsenden. Aktives Modul: Echolink. Um Hilfe zu erhalten bitte Null Raute senden.”

Hilfe-Modul

SvxLink verfügt über eine eingebaute Hilfe – Funktion welche der Funkamateur zur Laufzeit aufrufen kann. Dies geschieht gewöhnlich durch das Senden des DTMF Kommandos 0# zur SvxLink Station. Dadurch bekommt das Modul Hilfe den Fokus (es wird aktiviert). Nach dem das Modul den Fokus hat werden alle, auf diesem Link / Relais vorhandenen Module mit deren Modul ID gesprochen. Durch das Senden der Modul-ID wird die Hilfe des jeweiligen Moduls gesprochen. Nach dem die Hilfe gesprochen wurde befindet man sich wieder im Hauptmenü des Hilfe Moduls. Durch das Senden eines leeren Kommandos (einzelne Raute [#]) wird das Modul beendet. In der Standard Konfiguration sollten folgende Module verfügbar sein.

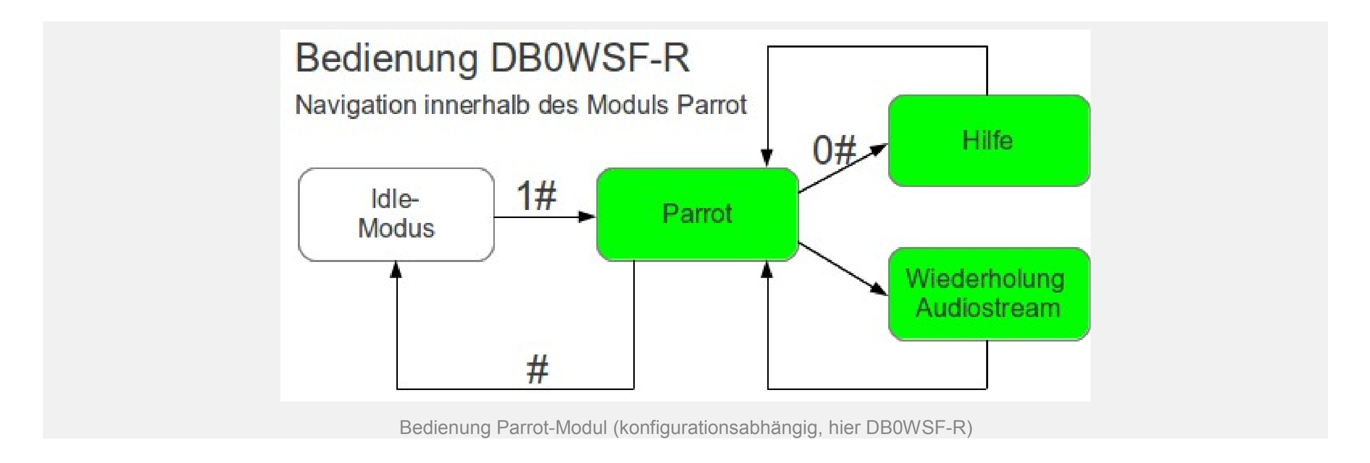
Modul ID	Modulanme
0	Hilfe
1	Sprachpapagei
2	Echo Link
3	Metar Information
4	Mailbox
5	Ausbreitungs Monitor
6	Selektiv Ruf (Encoder)



Bedienung Hilfe-Modul

Parrot-Modul (Sprachpapagei)

Dieses Modul speichert die empfangenen Signale und sendet sie erneut aus. Das Modul wird in der Standardkonfiguration mit dem Kommando 1# aktiviert und kann zur Überprüfung der eigenen Aussendung genutzt werden. Sollte die DTMF Erkennung nicht zu 100% Funktionieren kann mit Hilfe dieses Moduls festgestellt werden welche DTMF Zeichen vom SvxLink nicht erkannt werden. Das wird gemacht indem das Modul aktiviert wird und Anschließend eine DTMF Folge eingegeben wird. Nachdem diese DTMF Folge mit einer Raute abgeschlossen wurde werden die zuvor eingegebenen Zeichen wieder vorgelesen.

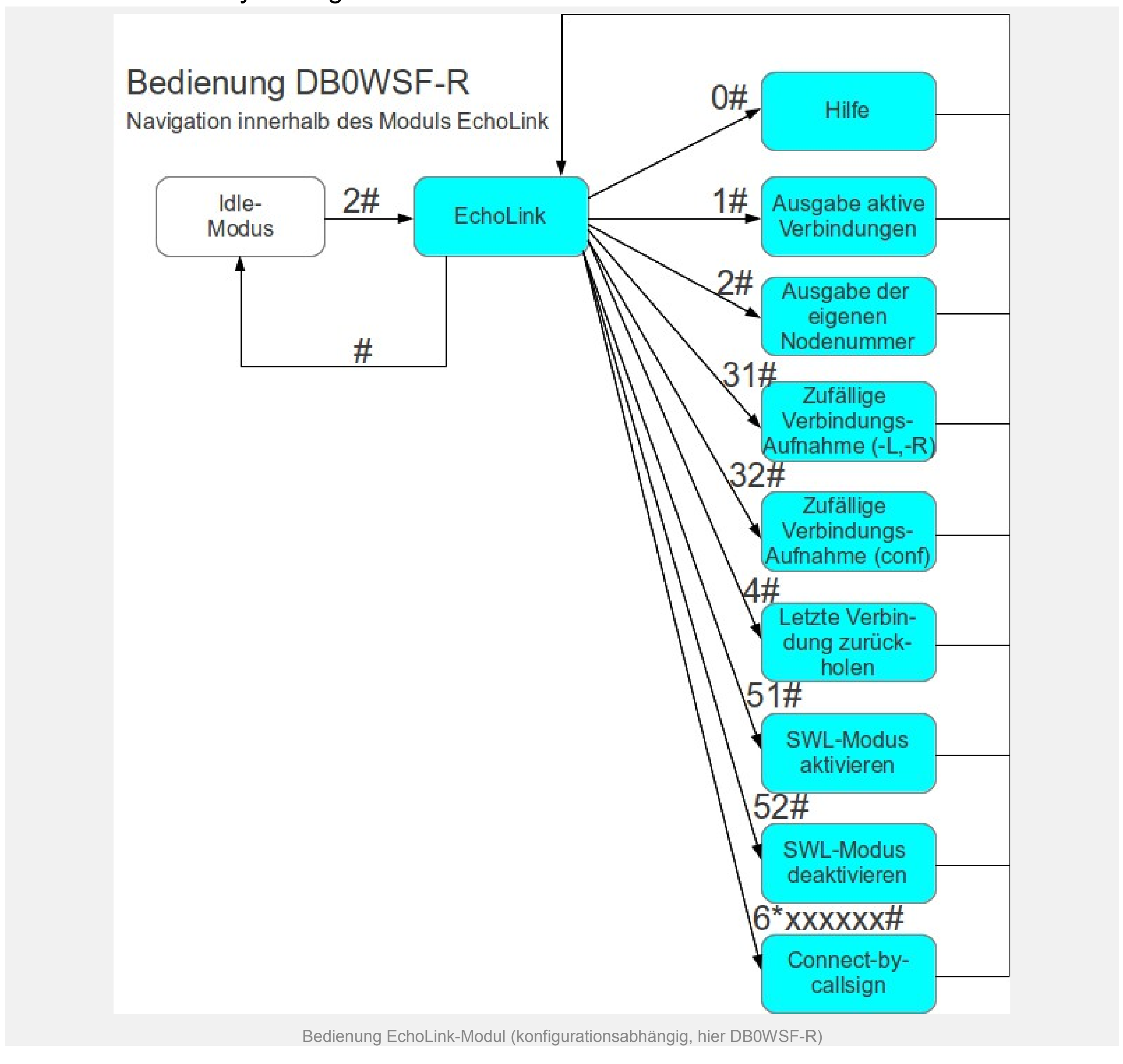


Bedienung Parrot-Modul (konfigurationsabhängig, hier DB0WSF-R)

Echolink-Modul

Mit diesem Modul ist es möglich Verbindungen zu anderen Echolink-Stationen herzustellen. Die Aktivierung erfolgt gewöhnlich durch das Senden des DTMF Kommandos 2#. Nach dem das Modul den Fokus hat stehen folgende Kommandos zur Verfügung.

Kommando	Funktion
0	Abspielen der Hilfe
1	Abfrage bestehender Verbindungen
2	Ansage der Lokalen Node ID
31	Verbindungsaufbau zu einem zufälligen Link oder Relais
32	Verbindungsaufbau zu einem zufälligen Konferenzserver
4	Letzte Verbindung zurückholen
50	SWL Modus Deaktivieren
51	SWL Modus Aktivieren
6*	Connect by Callsign



Bedienung EchoLink-Modul (konfigurationsabhängig, hier DB0WSF-R)

Connect-by-callsign

Die Connect by Callsign Funktion ist dazu gedacht eine Verbindung zu Stationen herzustellen von denen die Node ID unbekannt ist. Dabei ist wie folgt vorzugehen. Die Funktion ist mit 6* einzuleiten danach ist die Buchstaben-Zahlen-Kombinantion des Rufzeichens einzugeben. Jedes Zeichen ist nur einmal zu senden, ähnlich der intelligenten SMS-Texteingabe bei Mobiltelefonen. Die Kombination kann aus untenstehender Tabelle abgelesen werden.

Taste	1	2	3
Bedeutung	1	A B C 2	D E F 3
Taste	4	5	6
Bedeutung	G H I 4	J K L 5	M N O 6
Taste	7	8	9
Bedeutung	P Q R S 7	T U V 8	W X Y Z 9
Taste	*	0	#
Bedeutung	KEINE	0	KEINE

Wird nach der Eingabe des Rufzeichens oder eines Teils des Rufzeichens ein * gesendet wird dies als Wildcard ausgewertet. Diese funktioniert jedoch nur wenn der * das letzte Zeichen im Kommando ist. Hier einige Beispiele:

6*331262 gibt unter anderem DF1AMB zurück. Dabei werden DF1AMB-L und DF1AMB-R jedoch vollständig ingnoriert. Soll jedoch auch nach den anderen beiden Varianten von DF1AMB gesucht werden sollte der Suchbefehl 6*331262* lauten.

Hier noch weitere Beispiele:

Kommando	Station
6*760789*	SM0SVX*
6*36095*	DM0ZL*
6*320973*	DB0WSF*

Wichtig hierbei ist jedoch, dass es vorkommen kann das die Suche fehlschlägt. Dies ist auf zwei mögliche Uraschen zurückzuführen. Eine Ursache ist ein Fehler im eingegebenen Rufzeichen. Die andere Ursache ist, dass die gesuchte Station nicht verfügbar ist. Dies ist vergleichbar mit einem abgschaltenen Mobiltelefon.
next up previous contents

Mailbox-Modul

Im Leerlauf kann dieses Modul durch das Senden des Kommandos 4# aktiviert werden. Nach dem Aktivieren des Moduls wird man aufgefordert seine Nutzernummer gefolgt vom Passwort einzugeben. Das Login Kommando sieht z.B wie folgt aus:

1 2 3 1 2 3 4 5 6 7

Die ersten drei Ziffern entsprechen dabei immer der Nutzernummer und die letzten maximal 7 Ziffern entsprechen dem Passwort. Nach dem erfolgreichen Login kann man nun wählen ob man seine neuen Nachrichten hören möchte oder ob man eine neue Nachricht versenden möchte.

Mailbox: Nachrichten auslesen

Zum Lesen neuer Nachrichten ist nach erfolgereichem Login das Kommando 1# zu senden. Nachdem die Nachricht abgespielt wurde hat man die Möglichkeit diese Nachricht zu löschen oder dem Absender eine Antwort zu schicken und die Nachricht danach zu löschen oder Die Nachricht noch einmal abzuspielen.

Kommando	Aktion
1	Nachricht löschen
2	Nachricht beantworten und anschließend löschen
3	Nachricht erneut Abspielen

Mailbox: Nachricht versenden

Mit Hilfe des Komandos 2# kann man, nach erfolgreichem Login in die Mailbox, eine Nachricht verschicken.
Dazu muß man die Nutzernummer des Empfängers kennen. Man wir zu Beginn des Versendens aufgefordert die Nutzernummer des Empfängers einzugeben.
Sollte man jedoch eine Nachricht beantworten wollen ist es nicht notwendig die Nutznummer des Empfängers einzugeben.
Nach der Eingabe des Kommandos wird man aufgefordert einen kurzen Betreff anzugeben. Dieser darf in keinem Fall länger als 10 Sekunden sein.
Wurde der Betreff gesprochen, wird man aufgefordert die Nachricht zu sprechen. Auch für die Nachricht besteht eine Zeitbegrezung. Diese beträgt im Normalfall 2 Minuten. Jedoch kann diese Einstellung auch vom Betreiber des SvxLinks-Nodes je nach Bedarf geändert werden.

Mailbox: Nachschauen, ob Nachrichten bereitliegen

Durch eine Zusatzfunktion ist es möglich auch ohne Login in die Mailbox die Anzahl der wartenden Nachrichten für einen Nutzer abzufragen. Dabei ist es ebenfalls egal ob ein anderes Modul aktiv ist oder ob SvxLink sich im Leerlauf befindet. Der zu verwendende Befehl hierfür sieht wie folgt aus:

*4 123 #

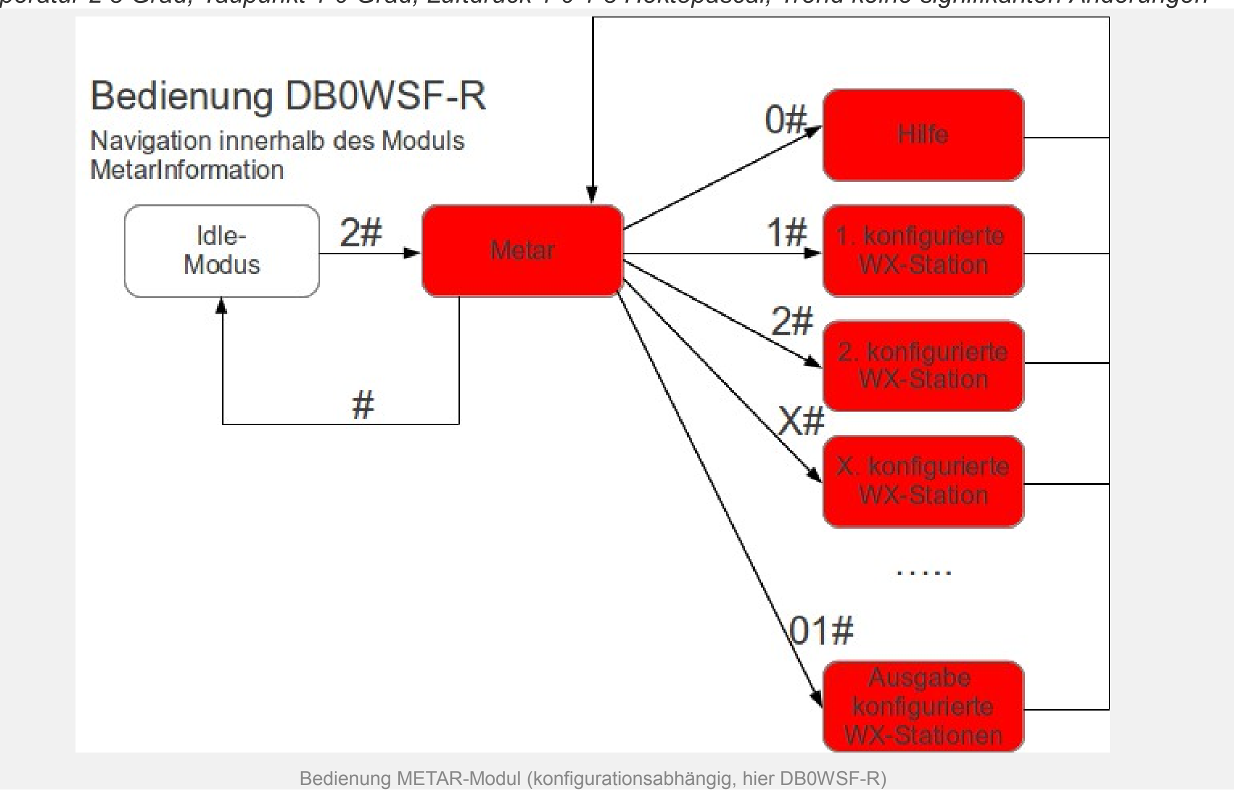
Der erste Teil ist eine “`Vorwahl”‘ um SvxLink mitzuteilen, dass der folgende Befehl von dem Modul nummer 4, also in diesem Fall die Mailbox, verarbeitet werden soll. Der zweite Fett geschriebene Teil ist die Nutzernummer.
Die Antwort von SvxLink sieht dann wie folgt aus:

Delta Foxtrott Eins Alpha Mike Bravo es sind Sechs neue Nachrichten vorhanden.

METAR-Modul

Mit Metar Information ist es möglich den aktuellen Wetterbericht von Wetterstationen abzurufen die sich im Allgemeinen an oder in der Nähe von internationalen Verkehrsflughäfen befinden. Dazu muss das Modul durch das Senden des Kommandos 3# aktiviert werden. Nach dem Aktivieren kann der Wetterbericht von vordefinierten Stationen durch das Senden der Stations ID abgerufen werden. Es ist ebenfalls möglich sich die Liste der vordefinierten Stationen ansagen zu lassen. Dies geschieht durch das Senden des Kommandos 01#. Es ist ebenfalls möglich die Metarinfos von nicht vordefinierten Flughäfen abzurufen. Hierfür muss die ICAO Kennung des Flughafens bekannt sein. Die Verwendung der Suchfunktion wird später beschrieben.

Die Ausgabe der Wettermeldungen erfolgt im METAR-Stil bzw. ähnlich der ATIS an Verkehrsflughäfen.
Hier ein Beispiel:
"*Leipzig-Halle Flughafen, Met-Report Zeit 1 0 5 0, Wind 2 4 0 Grad mit 1 2 Knoten, Sichtweite mehr als 1 0 km, Temperatur 2 3 Grad, Taupunkt 1 0 Grad, Luftdruck 1 0 1 3 Hektopascal, Trend keine signifikanten Änderungen*"



Bedienung METAR-Modul (konfigurationsabhängig, hier DB0WSF-R)

HINWEIS: Die Daten werden erst unmittelbar nach der Eingabe des Kommandos vom NOAA-Server abgerufen. In Abhängigkeit der Netz- und Serverbelastung kann es zu Verzögerungen oder Timeouts kommen, so dass die Ansage verzögert oder gar nicht erfolgt. In diesem Fall einfach später nochmals versuchen.

METAR: Suchfunktion 1

Bei dieser Suchfunktion wird die ICAO Kennung im SMS-Stil eingegeben und die einzelnen Zeichen werden mit * voneinander getrennt. Die Ziffer welche dem jeweiligen Zeichen zugeordent ist muss dabei so oft gesendet werden bis das Zeichen erreicht ist (SMS-Eingabe bei sehr alten Mobiltelefonen ohne automatische Texterkennung).

Taste	1	2	3
Bedeutung	KEINE	A B C	D E F
Taste	4	5	6
Bedeutung	G H I	J K L	M N O
Taste	7	8	9
Bedeutung	P Q R S	T U V	W X Y Z
Taste	*	0	#
Bedeutung	Trenner	KEINE	KEINE

Die Methode soll hier an zwei kurzen Bespielen erklärt werden. Der Flughafen Memmingen hat die ICAO Kennung E D J A. Aus der oben stehenden Tabelle kann nun die Kombination entnommen werden. Diese lautet 33 * 3 * 6 * 2. Für den Flughafen Zürich (L S Z H) lautet der Suchbefehl 555 * 7777 * 9999 * 44.

METAR: Suchfunktion 2

Diese Suchfunktion funktioniert ähnlich wie die erste jedoch ist hier kein Trennzeichen notwendig um die Zeichen voneinander zu trennen. Bei dieser Suchmethode wird zuerst die Taste angegeben auf welcher das Zeichen der zu finden ist, danach wird die Stelle des Zeichens angegeben. Die folgende Tabelle soll das veranschaulichen.

Taste	1	2	3
Position	1 2 3	1 2 3	1 2 3
Zeichen	A B C	D E F	
Taste	4	5	6
Position	1 2 3	1 2 3	1 2 3
Zeichen	G H I	J K L	M N O
Taste	7	8	9
Position	1 2 3 4	1 2 3	1 2 3 4
Zeichen	P Q R S	T U V	W X Y Z
Taste	*	0	#
Position	1 2 3	1 2 3	1 2 3
Zeichen			

Hier noch ein Beispiel zur Verwendung dieser Suchmethode. Als Beispiel dient wieder der Flughafen Memmingen (E D J A). Das einzugebende Kommando lautet jetzt also 32 31 51 21.

Selektivruf-Modul

Mit diesem Modul ist es möglich, 18 verschiedene Arten von Selektivrufen zu senden. Dies geschieht nach dem Aktivieren des Moduls durch die Eingabe der Selektivrufart und der zu rufenden Nummer.
Somit ist das Kommando zweigeteilt, ähnlich wie das Login Kommando der Mailbox. Hier repräsentieren die ersten zwei Zeichen den Selektivruftyp und die restlichen maximal 18 Zeichen werden als Selektivruf in der gewählten Variante ausgestrahlt. In der folgenden Tabelle sind die Selektivruftypen und deren Selektivruf ID zu finden.

Selektivruf ID	Selektruf Typ	Tonlänge
01	ZVEI1	70ms
02	ZVEI2	70ms
03	ZVEI3	70ms
04	PZVEI	70ms
05	DZVEI	70ms
06	EEA	40ms
07	CCIR1	100ms
08	CCIR2	70ms
09	VDEW	100ms
10	CCITT	100ms
11	NATEL	70ms
12	EIA	33ms
13	EURO	100ms
14	MODAT	40ms
15	PDZVEI	70ms
16	PCCIR	100ms
17	AUTO-A	70ms
18	QC2	1 / 3s

In der Standard Konfiguration hat dieses Modul die Nummer 6.
Selektivrufe können auch abgesetzt werden wenn ein anderes Modul aktiv ist. Hierfür ist es notwendig SvxLink mitzuteilen, dass das folgende Kommando nicht für das aktive Modul bestimmt ist sondern für ein anderes Modul. Dies geschieht mit Hilfe einer “Vorwahl” und sieht wie folgt aus:
***6** 01 03423 #
Die “Vorwahl” ist in diesem Beispiel **fett** dargestellt, das zu verwendende Verfahren Kursiv und die zu rufende Nummer ist untersrichen.
Es ist auch möglich mehrere Stationen der gleichen Nutzergruppe zu rufen. Dies geschieht mit Hilfe des Sterns. Um beispielsweise alle Nutzer der Gruppe 0342* zu rufen ist folgender Befehl zu verwenden:
(*6) 01 0342*

Selektivruf-Dekoder

Dies ist kein Modul im eigentlichen Sinne sondern nur eine Möglichkeit, SvxLink parallel zu bekannten DTMF-Funktionen per Mehrtonrufverfahren zu steuern.
Das verwendete Verfahren ist abhängig von der Konfiguration des SvxLinks. Somit kann man auch ohne DTMF-Encoder bestimmte Aktionen auslösen, dies ist interessant für Funkamateure mit kommerziellen Funkgeräten ohne eingebauten DTMF-Encoder. Da dieses Mehrtonrufverfahren auch zum selektiven Anruf einzelner Amateurfunkstationen verwendet werden kann, kann natürlich nur ein bestimmter Selektivrufbereich genutzt werden. Die Empfangenen Befehle werden dabei in Makrofaufrufe von SvxLink umgewandelt.

Die Konfiguration erfolgt in der /etc/svxlink/svxlink.conf. Es müsen die Variablen SEL5_MACRO_RANGE, SEL5_DET, SEL5_DEC_TYPE sowie SEL5_TYPE konfiguriert werden.

Hier nun ein Beispiel:

Selektivrufnr (z.B. ZVEI-1)	SvxLink-Macro (DTMF)	Aktion
03451	D03451#	MetarInfo von Leipzig
03452	D03452#	MetarInfo von Stuttgart
03453	D03453#	MetarInfo von München
03454	D03454#	MetarInfo von Köln/Bonn

Propagationmonitor-Modul

Dieses Modul kann zwar aktiviert werden aber es führt keine Funktionen aus. Es läuft im Normalfall im Hintergrund und empfängt DX Meldungen aus dem Internet per Email. Sobald eine solche Meldung empfangen wurde, wird diese von SvxLink angesagt.
Beispielansagen für den Ausbreitungsmonitor:

“Sporadic-E Öffnung auf 6m, die maximal nutzbare Frequenz ist 68MHz über JN60"
“*Transäquatorialöffnung auf 6 m”*

Module

DTMF-Repeater

Wenn dieses Modul aktiviert ist, sendet es alle empfangenen DTMF-Zeichen wieder aus. Dabei werden bis auf das #-Symbol alle DTMF-Zeichen bzw. Kommandos vom SvxLink-Kernsystem ignoriert. Das #-Symbol deaktiviert das DtmfRelais-Modul

Dieses Modul wird üblicherweise bei einem SvxLink-Relais genutzt um andere Dienste fernzusteuern. Die DTMF-Stummschaltung sollte dabei aktiviert sein, damit die DTMF-Töne nicht über das Relais ausgesendet werden.

Aktivierung
in der /etc/svxlink/svxlink.conf durch Hinzufügen des Eintrages

[RepeaterLogic]

...

MODULES=...,ModuleDtmfRepeater,...

...

Beispielkonfiguration in der /etc/svxlink/svxlink.d/ModuleDtmfRepeater.conf

[ModuleDtmfRepeater]

NAME=DtmfRepeater

PLUGIN_NAME=DtmfRepeater

ID=7

TIMEOUT=5

REPEAT_DELAY=500

Beschreibung

NAME

Name des Moduls

ID

Identifikationsnummer für dieses Modul. Diese Identifikationsnummer darf im gesamten System nur einmalig vergeben werden.

TIMEOUT

Zeit in Sekunden, nach der das Modul automaisch deaktiviert wird.

REPEAT_DELAY

Pausenzeit in Millisekunden, nach der die empfangenen Informationen wieder ausgesendet werden.

Echolink

Das Modul “Echolink” ermöglicht die Verbindung mit anderen Echolink-Stationen über das Internet.
Das Echolink-Modul wird genutzt um eine Verbindung mit anderen Echolinkstationen herzustellen. Um eine Verbindung herzustellen geben Sie einfach die Node-Nummer der Station gefolgt von einer Raute (#) ein. Um die Verbindung wieder zu trennen senden Sie eine einzelne Raute (#).
Um das Echolink-Modul wieder zu verlassen drücken Sie erneut #.

Es gibt eine Reihe von Unterkommandos, die innerhalb des Echolink-Modules genutzt werden können.

1# – Zeigt alle verbundenen Stationen
2# – Ausgabe der lokalen Echolink Nodenummer
31# – Verbindung mit einen zufällig gewählten Link oder Repeater
32# – Verbindung mit einer zufälligen Konferenz
4# – mit der Station wiederverbinden, die zuletzt getrennt wurde
50# – nur RX-Modus deaktivieren
51# – nur RX-Modus aktivieren

Die Funktion “per Rufzeichen verbinden” ermöglicht es die Verbindung zu einer Station herzustellen, deren Nodenummer nicht bekannt ist. Dabei werden die Rufzeichen werden auf Ziffern nach folgender Methode abgebildet:

ABC=2, DEF=3, GHI=4, JKL=5, MNO=6, PQRS=7, TUV=8, WXYZ=9. Es handelt sich um die selbe Abbildung wie bei den meisten Telefonen.

Buchstaben werden auf die zugehörigen Zeichen abgebildet und diese entsprechen den zugehörigen Nummern. Alle anderen Zeichen werden der Ziffer 1 zugeordnet. Ein Such-Kommando, beginnt mit einem Stern, danach folgt der Rufzeichen-Code. Wenn Sie also nach dem Rufzeichen SM3SVX-L suchen so senden Sie “*76378915#”;

Da die Codes nicht eindeutig sind, wird Ihnen eine Liste gefundener Stationen angeboten, die dem gesuchten Muster entsprechen und aus denen Sie auswählen können. Das Stern-Zeichen entspricht dabei einer Art Joker, d.h. ein beliebiges Zeichen. Wenn Sie beispielsweise alle Stationen mit der Zeichenfolge “SM3″ suchen, dann senden Sie einfach “*763*#”.

Für weiter Informationen zum Echolinksystem schauen Sie bitte auf die EchoLink-Webseite http://www.echolink.org.
Aktivierung
in der /etc/svxlink/svxlink.conf durch Hinzufügen des Eintrages

```
MODULE=...,ModuleEchoLink,...
```

Konfigurationsdatei /etc/svxlink/svxlink.d/ModuleEchoLink.conf

```
[ModuleEchoLink]
NAME=EchoLink
ID=2
TIMEOUT=60
ALLOW_IP=192.168.0.0/24
#DROP_INCOMING=^(.*)$
#REJECT_INCOMING=^(.*)$
#ACCEPT_INCOMING=^(.*)$
#REJECT_OUTGOING=^(.*)$
#ACCEPT_OUTGOING=^(.*)$
#CHECK_NR_CONNECTS=3,300,120
```

```
SERVERS=europe.echolink.org servers.echolink.org
CALLSIGN=DB0ABC-R
PASSWORD=*******
SYSOPNAME=Your name
LOCATION=[Svx] City
#PROXY_SERVER=
#PROXY_PORT=
#PROXY_PASSWORD=
#BIND_IP=192.168.0.10
#REJECT_CONF=1
USE_GSM_ONLY=1
MAX_QSOS=6
MAX_CONNECTIONS=7
#AUTOCON_ECHOLINK_ID=
#AUTOCON_TIME=
#LINK_IDLE_TIMEOUT=60
LOCATION_INFO=LocationInfo
DESCRIPTION="You have connected to a SvxLink node,\n"
"a voice services system for Linux with EchoLink\n"
"support. For more info check out:\n"
"http://sourceforge.net/apps/trac/svxlink/ \n"
"\n"
"Locator: AB12cd\n"
"QRG : 438.700 MHz (-7.6MHz)\n"
"CTCSS : NONE\n"
"Antenna: X50\n"
"Output : 15W EIRP\n"
```

Beschreibung

NAME

Name des Moduls

ID

Die eindeutige ID unter der das Modul aktiviert wird

TIMEOUT

Zeit in Sekunden, nach der das Modul deaktiviert wird

ALLOW_IP

Netzwerkadresse aus der ein lekaler Connect erlaubt wird, z.B. ALLOW_IP=192.168.0.0/24

DROP_INCOMING

Regex-Syntax, EchoLink-Stationen deren Connect-Anfragen ignoriert werden (keine Information an die Gegenstation)

REJECT_INCOMING

Regex-Syntax, EchoLink-Stationen, deren Connectanfragen abgewiesen werden (Information an die Gegenstation)

ACCEPT_INCOMING

Regex-Syntax, EchoLink-Stationen, deren Connectanfragen akzeptiert werden.

ACCEPT_OUTGOING

Regex-Syntax, wenn definiert, dann werden nur zu den Stationen aufgebaut, die der Regex-Definition entsprechen

REJECT_OUTGOING

Regex-Syntax, wenn definiert wird der (ausgehende) Verbindungsaufbau unterbunden.

REJECT_CONF

Wenn =1, dann werden alle eingehenden Verbindung deren Namen CONF am Ende enthalten abgewiesen. Dabei wird an die anfragende Station eine Information gesendet, dass die Verbindung nicht aufgebaut wird. Es

werden alle Stationen im Konferenz-Modus geblockt. Verbindet sich eine Station mit einer Konferenz nach dem die Verbindung zur eigenen Station bereits aufgebaut wurde, so wird die Verbindung zu dieser Station unterbrochen (disconnected).

CHECK_NR_CONNECTS

Wenn eine Station zu oft connected, so kann die Anzahl der Versuche begrenzt werden, ein erneuter Connectversuch wird dann abgewiesen. Beispiel CHECK_NR_CONNECTS=3,300,120
Baut eine Station 3x innerhalb von 300 Sekunden eine Verbindung zur eigenen Station auf, so wird diese Station für 120Minuten gesperrt

SERVERS

Leerzeichenseparierte Liste von Echolink-Servern. Wird die Verbindung zum Ersten getrennt, wird automatisch mit dem nächsten EchoLink-Server verbunden:
SERVERS=europe.echolink.org asia.echolink.org

CALLSIGN

Das Rufzeichen unter dem der eigene Node arbeitet (-L = Link, -R = Repeater), z.B. CALLSIGN=DB0ABC-R

PASSWORD

Das Passwort für die Anmeldung beim EchoLink-Verzeichnisdienst im Klartext(!)

SYSOPNAME

Der Name des Sysops

USE_GSM_ONLY

Auf Systemen mit schwacher CPU kann es sinnvoll sein, die automatische Codec-Umschaltung (von GSM nach SPEEX) zu verhindern, die standardmäßig zwischen SvxLink-Nodes ausgeführt wird. SPEEX benötigt eine höhere Rechenleistung, die auf schwachen Systemen unter Umständen zu unterbrochener Audioausgabe führen kann.

LOCATION

Die Name der Stadt/Dorfes wo sich der Link bzw. Repeater befindet, max. 27 Zeichen

MAX_QSOS

Die maximale Anzahl gleichzeitiger Verbindungen

MAX_CONNECTIONS

Wenn mehr Stationen als MAX_QSOS eine Verbindung aufbauen, so wird die Verbindung nur angenommen um der Gegenseite mitzuteilen, dass die Verbindungsabfrage auf Grund zu hoher QSO-Teilnehmerzahl abgelehnt wird. Bei mehr Anfragen als in MAX_CONNECTIONS definiert werden die Anfragen ignoriert.

PROXY_SERVER

Wenn gesetzt, erfolgt der Verbindungsaufbau zum konfigurierten EchoLink-Proxyserver. Aus- und eingehende Verbindungen werden ab sofort darüber geleitet.

PROXY_PASSWORD

Das Passwort für den EchoLink-Proxyserver

PROXY_PORT

Der Netzwerkport über den der Proxyserver erreicht wird.

BIND_IP

Mit diese Konfigurationsvariable kann der EchoLink-UDP-Socket an eine spezielle IP-Adresse/-Interface gebunden werden. Diese Konfigurtion kann sinnvoll sein, wenn der Computer mehrere Netzwerkinterfaces besitzt.
Beispiel: BIND_IP=192.168.0.10

AUTOCON_ECHOLINK_ID

Konfigurieren Sie hier eine EchoLink-Nodenummer des EchoLink-Nodes zu dem eine Verbindung automatisch aufgebaut werden soll.

AUTOCON_TIME

Zeit in Sekunden zwischen zwei Verbindungsversuchen zum gewünschten EchoLink-Node

LINK_IDLE_TIMEOUT

Zeit, nach der die Verbindung zur Gegenstation abgebaut wird, wenn keine Aktivität feststellbar ist.

DESCRIPTION

Eine längere Beschreibung, die der Gegenseite Informationen zur eigenen Station übermitteln, z.B. CTCSS, Antenne, TRX, …

Hilfe

Das Hilfemodul ermöglicht es Hilfe zum Systam als Ganzes zu bekommen. Im Moment steht die Hilfe allerdings nur in englischer Sprache zur Verfügung.
Senden Sie die ID-Nummer des Systemen zu dem Sie eine Hilfe erhalten wollen und bestätigen Sie wie immer mit der Nummerntaste (#)

Aktivierung
in der /etc/svxlink/svxlink.conf durch Hinzufügen des Eintrages

```
[RepeaterLogic]
...
MODULES=...,ModuleHelp,...
...
```

Beispielkonfiguration in der /etc/svxlink/svxlink.d/ModuleHelp.conf

```
[ModuleHelp]
NAME=Help
ID=0
TIMEOUT=60
```

Beschreibung

NAME
Name des Moduls

ID
Identifikationsnummer für dieses Modul. Diese Identifikationsnummer darf im gesamten System nur einmalig vergeben werden.

TIMEOUT
Zeit in Sekunden, nach der das Modul automaisch deaktiviert wird.

METAR

Das METAR-Modul ermöglicht das Abrufen von aktuellen Wetterinformationen, i.A. von Wetterstationen, die sich an oder in der Nähe von internationalen Verkehrsflughäfen befinden entsprechend der Konfiguration des SvxLink-Administrators.

Kommando Funktion
0# Hilfe allgemein
01# Aufzählung konfigurierter Wetterstationen
1# 1. konfigurierte Wetterstation
2# 2. konfigurierte Wetterstation
3# 3. konfigurierte Wetterstation
Xx# Xx. Wetterstation
Für weitere Informationen zu METARs siehe Metar bei Wikipedia.
Aktivierung in der /etc/svxkink/svxlink.conf durch

```
[RepeaterLogic]
...
MODULE=...,ModuleMetarInfo,...
...
```

Konfigurationsdatei /etc/svxlink/svxlink.d/ModuleMetarInfo.conf

```
[ModuleMetarInfo]
NAME=MetarInfo
ID=3
TIMEOUT=60
#STARTDEFAULT=EDDP
LONGMESSAGES=1
REMARKS=1
DEBUG=1
# insert ICAO airport shortcuts here. You can
# request the METAR by sending dtmf commands as follows
#
# 1# 2# 3# 4# 5# 6# 7# 8# 9# 10# 11# 12# 13# 14# 15# 16# 17# 18# 19# 20#
AIRPORTS=EDDP,EDDS,EDDM,EDDK,EDDV,LEMD,ESSB,KJAC,KTOL,ZBAA,KJFK,SKBO,SKSM,EDJA,ZMUB,EGLL,LEPA,EDDH,LIRF,
LIRN
```

Beschreibung

NAME

Der Name dieses Modules, z.B. *NAME=MetarInfo*

ID

Die eindeutige Identifikationsnummer, die für jedes SvxLink-Modul eindeutig vergeben werden muss. Die Aktivierung des Modules erfogt bei *ID=3* mit dem Kommando *3#*

TIMEOUT

Das Modul beendet sich nach der konfigurierten Zeit (in Sekunden) selbsständig, wenn keine Aktivität erfolgte.

STARTDEFAULT

Beim Aktivieren des METAR-Modules werden die Wetterdaten der konfigurierten Station sofort abgerufen und ausgegeben.

LONGMESSAGES

Bei bestimmten METAR-Informationen wird nicht die Kurzform ausgegeben sondern die vollständige Beschreibung des Tags, z.B. statt "NOSIG" -> "no significant change" oder "CAVOK" -> "ceiling and visibility OK" Für weitere Informationen schauen Sie bei Wikipedia unter METAR.

REMARKS

Wenn "=1", dann werden auch die METAR-Abschnitte im RMK (Remark)-Bereich ausgegeben. Hier sind u.a. Landebahninformationen, Wettertrends usw. zu finden. Leider gibt es hier keinen durchgängigen Standard, der

von allen Stationen eingehalten wird. Deshalb hier keine Garantie, dass jedes Tag auch ordnungsgemäß ausgegeben wird.
DEBUG
Wenn “=1″, dann erfolgt ein erweitertes Loggen im svxlink-Log, so können Fehler beim Abruf der Wetterinformationen analysiert werden.
AIRPORTS
Kommaseparierte Liste von Wetterstationen, die per Kommando abgerufen werden können. Der Abruf erfolgt in der Reihenfolge der Konfiguration. z.B.
AIRPORTS=EDDP,EDDM,EDDS,EDDH
Bei dieser Konfiguration kann nach Aktivierung des METAR-Moduls und dem Kommando “1#” die Wetterinformation von Leipzig abgerufen werden, bei “2#” die von München, bei “3#” von Stuttgart usw…

Parrot

Der Sprachpapagei

Das Parrot-Modul gibt alles wieder was es selbst hört. Es kann als Simplex-Repeater genutzt werden oder um einfach zu prüfen, wie sich die eigene Aussendung über Funk anhört. Weiterhin sagt es die DTMF-Zeichen an, die man selbst gerade gesendet hat. Wie immer sind alle Eingaben durch das Nummernsymbol zu bestätigen. Dieses Modul wird durch das Senden eines einzelnen Nummern-Symbols (#) wieder verlassen.

Aktivierung
in der /etc/svxlink/svxlink.conf durch

```
[RepeaterLogic]
...
MODULES=...,ModuleParrot,...
...
```

Beispielkonfiguration in der /etc/svxlink/svxlink.d/ModuleParrot.conf

```
[ModuleParrot]
NAME=Parrot
ID=1
TIMEOUT=60
FIFO_LEN=60
REPEAT_DELAY=1000
```

Beschreibung
NAME
Name des Moduls
ID
Identifikationsnummer für dieses Modul. Diese Identifikationsnummer darf im gesamten System nur einmalig vergeben werden.
TIMEOUT
Zeit in Sekunden, nach der das Modul automatisch deaktiviert wird.
REPEAT_DELAY
Zeit in Millisekunden nach der der gespeicherte Audiostream wieder ausgegeben wird.

Propagation-Monitor

Der Ausbreitungsmonitor informiert über besondere Ausbreitungsbedingungen im VHF, UHF und SHF-Bereich. Die Aktivierung erfolgt dabei automatisch immer genau dann wenn besondere Ausbreitungsbedingungen gemeldet wurden (DX-Robot, VHF-DX, …)
Dieses Modul kann zwar aktiviert werden, besitzt aber keine weitere Funktionalität.

Aktivierung
in der /etc/svxlink/svxlink.conf durch Hinzufügen des Eintrages

```
[RepeaterLogic]
...
MODULES=...,ModulePropagationMonitor,...
...
```

Beispielkonfiguration in der /etc/svxlink/svxlink.d/ModulePropagationMonitor.conf

```
[ModulePropagationMonitor]
NAME=PropagationMonitor
PLUGIN_NAME=Tcl
ID=5
TIMEOUT=10
SPOOL_DIR=/var/spool/svxlink/propagation_monitor
```

Beschreibung

NAME
Der Name dieses Modules, z.B. *NAME=PropagationMonitor*

ID
Die eindeutige Identifikationsnummer, die für jedes SvxLink-Modul eindeutig vergeben werden muss. Die Aktivierung des Modules erfogt bei *ID=5* mit dem Kommando *5#*

TIMEOUT
Das Modul beendet sich nach der konfigurierten Zeit (in Sekunden) selbsständig, wenn keine Aktivität erfolgte.

SPOOL_DIR
Verzeichnis in dem die bearbeitete Textdatei, die aus der eMail erstellt wurde, abgelegt wird.

Einrichtung des Mailservers zum Empfang der DX-Meldungen vom DX-Robot:

- Nutzer svxlink auf dem SvxLink-Server erstellen
- Datei *.procmailrc* im Verzeichnis /home/svxlink erstellen mit folgenden Einträgen

```
# This is the configuration file for procmail that is used to
# extract received propagation alerts into separate files.
# This is used by the PropagationMonitor SvxLink module.
LOGFILE=/home/svxlink/procmail.log
VERBOSE=ON
# Store received propagation alerts here
MAILDIR=/var/spool/svxlink/propagation_monitor
# Put e-mails from the DXrobot (gooddx.net) in the folder below
:0:
* ^From.*DXrobot
/var/spool/svxlink/propagation_monitor/dxrobot
# Put e-mails from VHFDX (vhfdx.net) in the folder below, due to
# the change of the domain name you can comment out
# the next lines
#:0:
#* ^From.*DX.Sherlock
#/var/spool/svxlink/propagation_monitor/vhfdx
# the new domain name is dxmaps.com
:0:
```

* ^From.*warnings@dxmaps.com

/var/spool/svxlink/propagation_monitor/vhfdx

:0:

* ^From.*@gooddx.*

/var/spool/svxlink/propagation_monitor/dxrobot

Send all other, non matching e-mails, to user root.

Replace "!root" with "/dev/null" to throw non-matching

e-mails away instead or comment both lines below out

to just let non-matching e-mails pass through.

:0:

/dev/null

Selektivrufencoder

Mit dem Selektivrufencoder ist es möglich per DTMF aus 17 verschiedenen Selektivrufverfahren auszuwählen und einen Selektivruf auszulösen (z.B. DTMF nach ZVEI-1, DTMF nach EIA, usw.).

Aktivierung

in der /etc/svxlink/svxlink.conf durch Hinzufügen des Eintrages

[RepeaterLogic]

...

MODULES=...,ModuleSelCallEnc,...

...

Beispielkonfiguration in der /etc/svxlink/svxlink.d/ModuleSelCallEnc.conf

[ModuleSelCallEnc]

NAME=SelCallEnc

PLUGIN_NAME=Tcl

ID=6

TIMEOUT=10

Beschreibung

NAME

Name des Moduls

ID

Identifikationsnummer für dieses Modul. Diese Identifikationsnummer darf im gesamten System nur einmalig vergeben werden.

TIMEOUT

Zeit in Sekunden, nach der das Modul automaisch deaktiviert wird.

Dateien und Verzeichnisse

/etc/svxlink/svxlink.conf

Das systemweite svxlink-Konfigurationsfile

~/.svxlink/svxlink.conf

Das userspezifisches svxlink-Konfigurationsfile

~/.svxlink/svxlink.d/ModuleSellCallEnc.conf

Das userspezifische Konfigurationsfile für das Modul Selektivrufencoder, abhängig von der CFG_DIR Kofigurationsvariable.

/etc/svxlink/svxlink.d/ModuleSellCallEnc.conf

Das systemweite Konfigurationsfile für das Modul Selektivrufencoder, abhängig von der CFG_DIR Kofigurationsvariable.

TclVoiceMail

Dieses Modul beinhaltet ein einfaches Sprachmailboxsystem welches von den lokalen Teilnehmern genutzt werden kann. Es ist im Moment noch nicht möglich, Sprachnachrichten zu den Nutzern anderer Echolink-Knoten zu senden. Um dieses Sprachmailboxsystem nutzen zu können, müssen Sie eine eindeutige Nutzernummer und ein Passwort haben. Nehmen Sie bitte hierzu Kontakt mit Ihrem Sysop auf.

Wenn das Modul aktiviert wurde (Standardeinstellung 3#) dann werden sie aufgefordert Ihre Nutzernummer und ein Passwort einzugeben. Die Nutzernummer ist immer 3 Ziffern lang, das Passwort zwischen einem und sieben Ziffern.
Geben Sie Ihre Nutzernummer und Ihr Passwort hintereinander ein, z.B. 123456# wenn Ihre Nutzernummer 123 und Ihr Passwort 456 ist.
Danach befinden Sie sich in einem Kontextmenü, das Sie duch das System führen wird. Eine einzelne Raute (#) beendet die aktuelle Operation.

Um eine Sprachnachricht aufzunehmen nutzen Sie das Kommando 2# .
Das System fragt Sie jetzt nach dem Empfänger der Nachricht. Sie können beides miteinander verbinden, indem Sie das Kommando und die Nutzernummer hintereinander eingeben. Zum Beispiel: Wenn eine Nachricht für den Nutzer 123 aufgezeichnet werden soll, geben Sie bitte die Folge 2123# ein.

Der empfohlene Weg um Nutzernummern zu vergeben ist, die “Telefon-Methode” zu nutzen. Viele Telefone besitzen Buchstaben auf dem Tastenfeld, z.B. 2=ABC, 3=DEF, 4=GHI, 5=JKL, 6=MNO, 7=PQRS, 8=TUV, 9=WXYZ.
Um beispielsweise das Rufzeichen SM0SVX zu kodieren, nutzen Sie bitte die letzten drei Buchstaben und bilden Sie diese auf die Zahlen ab. Dann erhalten Sie die Nutzernummer 789. Im Falle einer Doppelvergabe fügen Sie einfach eine 1 hinzu. Zum Beispiel AFC und ADB lassen sich beide als 232 abbilden. Eine mögliche Abbildung wäre dann AFC=232 und ADB=233. Wenn diese Methode genutzt wird um Nutzernummern zu kodieren, dann kann ein Teilnehmer auf die Nutzernummer eines anderen Teilnehmers schließen.

Es ist für den Sysop möglich für jeden Nutzer eine eMail-Adresse zuzuweisen.
In diesem Fall erhält der Nutzer eine eMail, falls eine neue Sprachnachricht für ihn hinterlegt ist.

Um dieses Modul zu deaktivieren, drücken Sie die Raute-Taste (#) wenn sie sich im Hauptmenü befinden.

<u>Aktivierung</u>
in der /etc/svxlink/svxlink.conf durch Hinzufügen des Eintrages

```
[RepeaterLogic]
...
MODULES=...,ModuleTclVoiceMail,...
...
```

<u>Beispielkonfiguration in der /etc/svxlink/svxlink.d/ModuleTclVoiceMail.conf</u>

```
[ModuleTclVoiceMail]
NAME=TclVoiceMail
PLUGIN_NAME=Tcl
ID=4
TIMEOUT=60
```

<u>Beschreibung</u>

NAME
Name des Moduls

ID
Identifikationsnummer für dieses Modul. Diese Identifikationsnummer darf im gesamten System nur einmalig vergeben werden.

TIMEOUT

Zeit in Sekunden, nach der das Modul automaisch deaktiviert wird.

Dateien und Verzeichnisse

/etc/svxlink/svxlink.conf

Das systemweite svxlink-Konfigurationsfile

~/.svxlink/svxlink.conf

Das userspezifisches svxlink-Konfigurationsfile

~/.svxlink/svxlink.d/ModuleTclVoiceMail.conf

Das userspezifische Konfigurationsfile für das Modul TclVoiceMail, abhängig von der CFG_DIR Kofigurationsvariable.

/etc/svxlink/TclVoiceMail.conf

Das systemweite Konfigurationsfile für das Modul TclVoiceMail.

/var/spool/svxlink/voice_mail

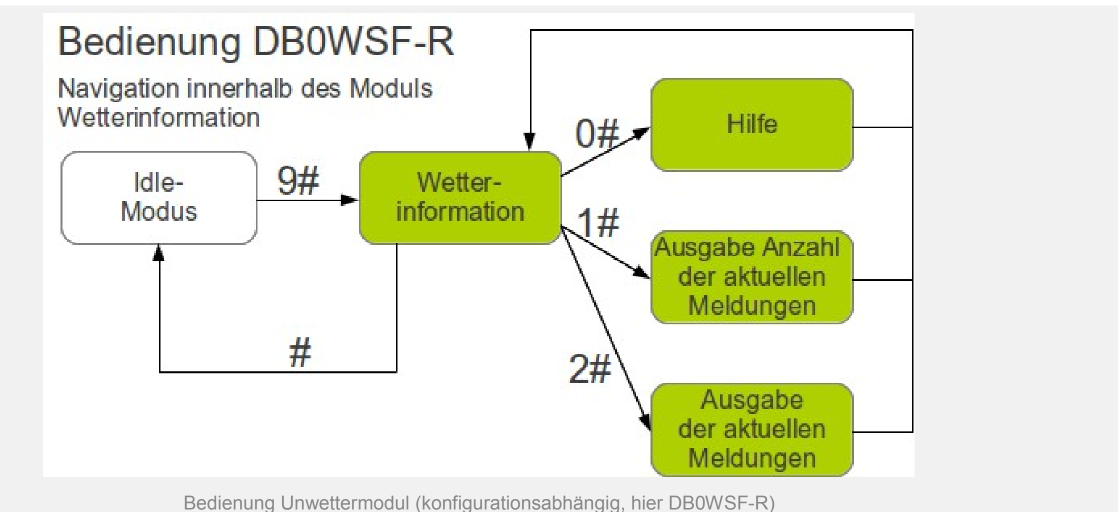
Spool-Verzeichnis für Sprachdateien (der User svxlink muß hier Schreibrechte haben!)

(Unwetterwarnung)

- Kein offizielles Modul -

Dieses Modul gibt eine Unwetterwarnung als Sprachausgabe aus, sobald diese per eMail empfangen wurde. Es wird seit einigen Monaten auf DB0WSF, DM0LEI und weiteren Repeatern bzw. Links getestet. Zum Betrieb ist ein Text-To-Speak-System notwendig, SvxLink selbst generiert keine dynamischen Sprachausgaben.
Zum Betrieb sind zwei TCL-Dateien notwendig, die hier heruntergeladen werden können:

WeatherInfo.tcl
ModuleWeatherInfo.tcl
Das File WeatherInfo.tcl nach */usr/share/svxlink/events.d* und ModuleWeatherInfo.tcl nach */usr/share/svxlink/modules.d* kopieren.
Weiterhin ist ein spool-Verzeichnis anzulegen: */var/spool/svxlink/weatherinfo*mit einer Schreibberechtigung für den User svxlink.



Bedienung Unwettermodul (konfigurationsabhängig, hier DB0WSF-R)

Nachfolgend Beispiele für Unwetterwarnungen:
Unwetterwarnung Landkreis Leipzig
Unwetterwarnung Landkreis Göttingen
Kommandos
0# – Hilfe
1# – Gibt die *Anzahl* der aktuellen Unwetterwarnungen aus
2# – Gibt die aktuellen Unwetterwarnungen in zeitlicher Reihenfolge aus
Beispielkonfiguration in der /etc/svxlink/svxlink.d/WeatherInfo.conf

```
[ModuleWeatherInfo]
NAME=WeatherInfo
PLUGIN_NAME=Tcl
ID=12
TIMEOUT=15
DELETE_AFTER=120
ALERT=0
SPOOL_DIR=/var/spool/svxlink/weatherinfo
PLAY_DIR=/usr/share/svxlink/sounds/de_DE/WeatherInfo
```

Beschreibung

NAME
Name des Moduls
ID
Identifikationsnummer für dieses Modul. Diese Identifikationsnummer darf im gesamten System nur einmalig vergeben werden.
TIMEOUT

Zeit in Sekunden, nach der das Modul automatisch deaktiviert wird.

DELETE_AFTER

Zeit in Minuten, nach der die gespeicherten Unwetterwarnungen automatisch gelöscht werden.

ALERT

=1, wenn vor der Unwetterwarnung ein Alarmton ausgegeben werden soll

SPOOL_DIR

Verzeichnis, in dem die Unwetterwarnungen erwartet werden, die Dateinamen entsprechen folgender Struktur:
CALL.Hashcode.wav – die auszugebende Wav-Datei
CALL.Hashcode.info – der Text aus dem die Unwetterwarnung generiert wurde

PLAY_DIR

Verzeichnis, in dem die Meldungen kopiert und verarbeitet werden. Nach der Ausgabe werden sie in das $PLAY_DIR/archive Verzeichns verschoben, in dem sie noch für DELETE_AFTER verbleiben um dann endgültig gelöscht zu werden.

(Verkehrsinformation)

- Kein offizielles Modul -

Dieses Modul (eigentlich “Gefahrenmeldung”) wird seit einigen Monaten getestet und funktioniert relativ gut. Im Release/Trunk wird es allerdings nie erscheinen, es soll nur demonstrieren was mit SvxLink möglich ist.

Es besteht auch eine begrenzte Steuerungsmöglichkeit nach Aktivierung des Modules.

Kommandos:
0# – Hilfe
1# – Gibt nur die *Anzahl* der aktuellen Verkehrsmeldungen aus
2# – Gibt alle aktuellen Verkehrsmeldungen in zeitlicher Reihenfolge aus

Beispielkonfiguration in der /etc/svxlink/svxlink.d/TrafficInfo.conf

```
[ModuleTrafficInfo]
NAME=TrafficInfo
PLUGIN_NAME=Tcl
ID=11
TIMEOUT=15
DELETE_AFTER=120
HINZ=1
SPOOL_DIR=/var/spool/svxlink/trafficinfo
PLAY_DIR=/usr/share/svxlink/sounds/de_DE/TrafficInfo
```

Beschreibung

NAME
Der Name des Moduls, z.B. NAME=TrafficInfo

PLUGIN_NAME

ID
Eindeutige ID des Moduls, mit der das Modul gestartet wird.

TIMEOUT
Zeit in Sekunden, nach der das Modul beendet wird, falls keine Aktivität feststellbar ist.

DELETE_AFTER
Zeit in Minuten, nach der die gespeicherten Verkehrsmeldungen automatisch gelöscht werden

HINZ
Ausgabe des Verkehrsmeldung-Aktivierungstones am Anfang und Ende der Meldung, siehe Hinz-Triller

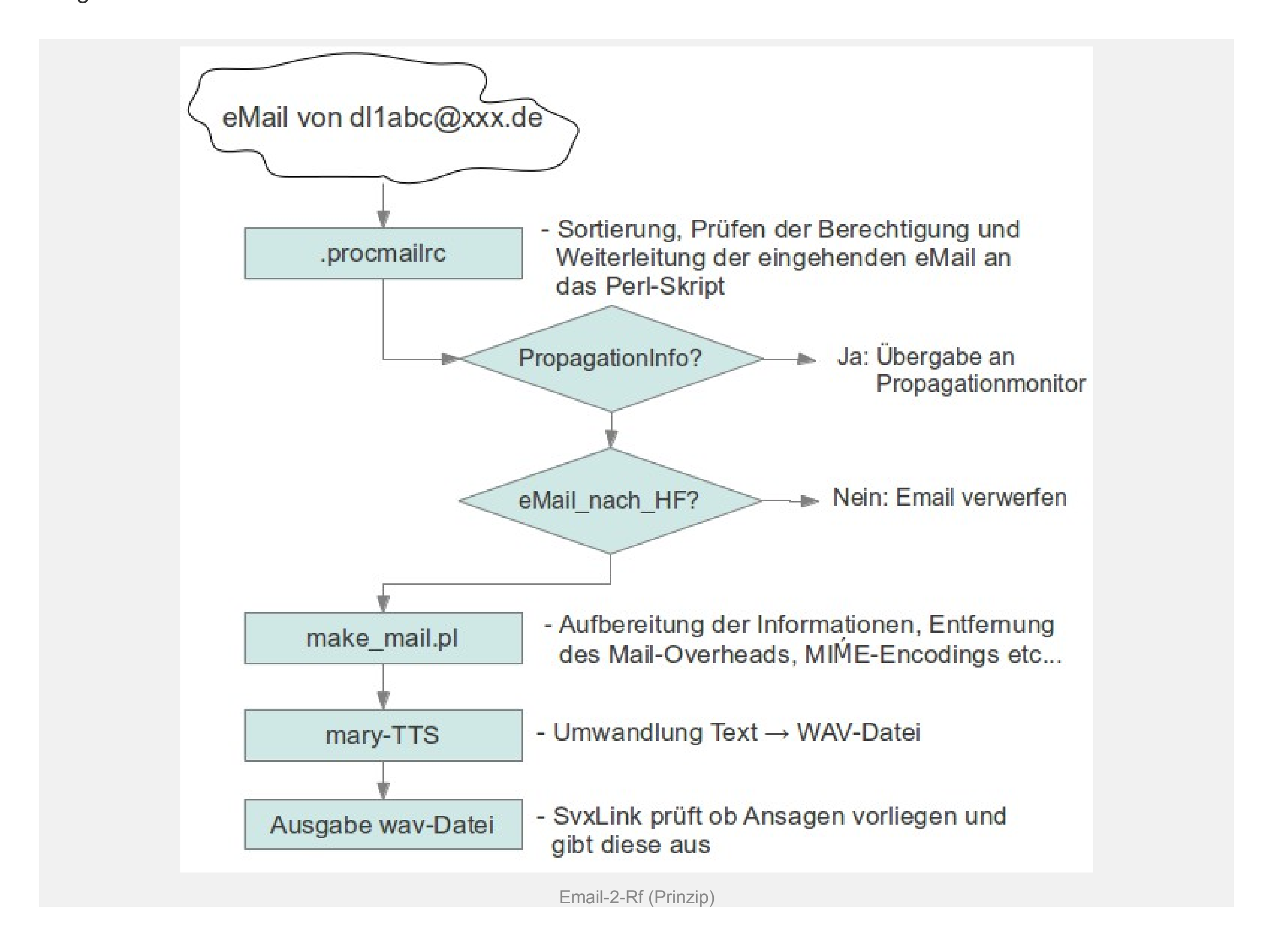
SPOOL_DIR
Verzeichnis, in der die auszugebenden Verkehrsmeldungen erwartet werden, die Dateien besitzen die Struktur:
CALL.Hashcode.wav – das auszugebende Wav-File
CALL.Hashcode.info – das .info-File einthält den Text aus dem das wav-file generiert wurde
Der Hashcode wird aus dem Text-Inhalt der Verkehrsmeldung generiert, CALL muß dem Rufzeichen entsprechen, unter dem der Repeater/Link konfiguriert wurde.

PLAY_DIR
Verzeichnis, in das die Files für die Ausgabe kopiert werden, nach dem Abspielen werden diese Files mit einem “old-” am Anfang des Dateinames versehen und in das*$PLAYDIR/archive*-Verzeichnis kopiert in dem sie für DELETE_AFTER verbleiben um danach gelöscht zu werden.

(eMail-2-Rf)

- Kein offizielles Modul -

Dieses Modul gibt einen als eMail empfangenen und in eine wav-Datei umgewandelten Text aus, es läuft seit einigen Monaten auf DB0WSF und DM0LEI.



Email-2-Rf (Prinzip)

Details

Download

Die Auto-Installations-CD von Martin (DF1AMB) kann man von hier herunterladen:

Download Iso-Image debian-svxlink-auto-install.iso Größe: 244416 kByte
ACHTUNG: Diese Installation läuft vollautomatisch ab, sie formatiert die 1.Festplatte und installiert ungefragt diese Distribution mit SvxLink! Bitte sicherstellen, dass auf dem Rechner keine wichtigen (ungesicherten) Daten vorhanden sind. Diese sind dann unwiederbringlich verloren! Die Installation erfolgt auf eigenes Risiko, eine Haftung wird nicht übernommen!

Weiterhin liegen hier ein paar SvxLink-Binaries für verschiedene Distributionen zum Download bereit: **Achtung, hierbei handelt es sich um eine neuere Trunk-Version mit neuen Features**.
Leider hat das Programm “checkinstall”, welches ich zum Erstellen der *.deb-Pakete verwende einen Bug, so dass es zu Problemen beim Installieren der *.deb-Pakete kommt, wenn vorher noch kein svxlink auf dem System installiert war. Abhilfe: Bitte das Skript
http://svxlink.de/download/svxlink-pre-deb.sh
als root auf den betreffenden Systemen vor der Installation des *.deb-Paketes starten.
Ubuntu:
https://launchpad.net/~felix.lechner/+archive/hamradio
svn-Version von Martin (DF1AMB), Rev. 2005:
32 bit-Version (svxlink_0.svn2005_i386.deb)
OpenSuse 11.3:
Für OpenSuSE 11.3 und 11.4 nutzen Sie bitte das hamradio-Repository, siehe Installation
Synthetische Mary-Soundfiles (deutsch, männlich und weiblich, verschiedene Sampleraten):
stimmen-mary.tar.bz2 (110.8MByte)
Entpacken Sie die Datei z.B. nach /tmp . diese Datei enthält verschiedene Unterverzeichnisse, z.B. 16k-female, 22.05k-male, usw…
Kopieren Sie das für Ihre svxlink-Version gültige Unterverzeichnis nach*/usr/share/svxlink/sounds* und benennen sie es nach *de_DE* um, so dass sich eine Struktur wie folgt ergibt:
/usr/share/svxlink/sounds/de_DE/Core
/usr/share/svxlink/sounds/de_DE/Default
/usr/share/svxlink/sounds/de_DE/DtmfRepeater
/usr/share/svxlink/sounds/de_DE/EchoLink
…
usw.

Diese Stimmen sind OpenSource-Stimmen (Mary, siehe http://mary.dfki.de/), welche qualitativ nicht optimal sind. Es ist auch ein qualitativ hochwertiger Sprachsatz vorhanden, dieser setzt allerdings das Vorhandensein einer kostenpflichtigen Lizenz voraus. Falls daran Interesse bestehen sollte, bitte Kontakt mit uns aufnehmen.
Weitere Erklärungen zu den Begriffen Release,Trunk und Branch können Sie auf derWikipedia-Seite nachlesen.

Installation

1) durch bereitgestellte Binärpakete
Einige Distributionen bieten fertig kompillierte Pakete für SvxLink an. Diese müssen Sie dann nur installieren.

Für OpenSuSE werden Binärpakete im hamradio-Repository bereitgestellt. An dieser Stelle herzlichen Dank an OM Walter (DL8FCL), der sich dieser Sache angenommen hat. Um SvxLink unter OpenSuSE 11.4 bis 13.2 zu installieren, fügen Sie einfach das entsprechende hamradio-Repository hinzu:

Repo für 11.4 http://download.opensuse.org/repositories/hamradio/openSUSE_11.4/
Repo für 12.1 http://download.opensuse.org/repositories/hamradio/openSUSE_12.1/
Repo für 12.2 http://download.opensuse.org/repositories/hamradio/openSUSE_12.2/
Repo für 12.3 http://download.opensuse.org/repositories/hamradio/openSUSE_12.3/
Repo für 13.1 http://download.opensuse.org/repositories/hamradio/openSUSE_13.1/
Repo für 13.2 http://download.opensuse.org/repositories/hamradio/openSUSE_13.2/
Dort finden sich auch noch mehr Programme für Funkamateure.

Für Ubuntu stellt Felix Lechner Pakete für verschiedene Ubuntu-Versionen zur Verfügung, siehe:
https://launchpad.net/~felix.lechner/+archive/hamradio
Um dieses Repository hinzuzufügen führen Sie aus:
sudo add-apt-repository ppa:felix.lechner/hamradio
Detailiertere Informationen zur Intallation von Software über ppa's erhalten Sie auf der Ubuntu-Webseite: http://wiki.ubuntuusers.de/Paketquellen_freischalten/PPA
2) oder für den Freak: "zu Fuß" durch Selberkompillieren
Achtung: Das Projekt ist zu Github gewechselt. Deshalb stehen nur da die neusten Versionen zur Verfügung!
Bitte auch hier unterscheiden, ob Sie ein Release installieren wollen oder einen Trunk. Weiterführende Informationen zu den Begriffen Release und Trunk finden Sie auf derWiki-Seite zur Versionsverwaltung.
Für beide Varianten gilt prinzipiell der selbe Ablauf, es unterscheidet sich nur die Herkunft der Quellen. Herunterladen können Sie sowohl den letzten Trunk als auch das Release der Einfachheit halber per svn. Installieren Sie zunächst *git* auf Ihren PC. Unter openSuSE per Yast odersudo zypper install git. Unter Ubuntu/Debian geben Sie ein sudo apt-get install git oder nutzen Sie das Software-Center.
- Herunterladen des Trunks
git clone https://github.com/sm0svx/svxlink.git
Es sind weitere Pakete erforderlich, unter Debian kann man diese mittels:
apt-get install gcc g++ make cmake libgcrypt-dev libgsm1-dev libsigc++-2.0-dev tcl-dev libspeex-dev libasound2-dev libpopt-dev libssl-dev libopus-dev groff
installieren. Eventuell gibt es Fehlermeldungen, falls einige Pakete bereits auf dem System installiert sind. Unter openSuSE kann die Installation mit yast oder zypper erfolgen, z.B.: zypper install gcc g++ make cmake libgcrypt-devel libgsm-devel libsigc++-2-devel tcl-devel libspeex-devel libopus-devel libasound2-devel libpopt-devel libssl-devel
Unter src finden Sie die c++-Dateien. Des weiteren sind die folgenden Schritte erforderlich:

```
cd svxlink/src

mkdir build

cd build

cmake -DUSE_QT=OFF -DCMAKE_INSTALL_PREFIX=/usr -DSYSCONF_INSTALL_DIR=/etc
-DLOCAL_STATE_DIR=/var -DCMAKE_BUILD_TYPE=Release ..

make

make install
```

Voraussetzung für die o.g. ist natürlich, dass sich auf Ihrem PC eine c++-Entwicklungsumgebung befindet. Probleme und Fehlermeldungen werden mit Sicherheit auftreten, da Sie am Anfang sicherlich noch nicht alle erforderlichen Pakete installiert haben.

SvxLink wurde von Tobias (SM0SVX) unter Fedora Linux entwickelt. Es sollte sich einfach unter Fedora installieren lassen, da hierfür Binärpakete zur Verfügung stehen. Da es unter Fedora entwickelt wurde ist es auch unter dieser Distribution am besten getestet worden.
SvxLink hat einige wenige abhängige Pakete. Die meisten sollten bereits auf einer normalen Linux-Arbeitsstation installiert sein. Qtel benötigt X/Qt aber der SvxLink-Server ist eine reine Konsolenapplikation.

SvxLink liefert einige RPM-Pakete aus:

libasync	notwendig. Hauptbestandteil
echolib	benötigt für EchoLink
svxlink-server	der SvxLink-Server
qtel	Das Qt EchoLink-Client-Programm
xxx-devel Pakete	Entwicklungspakete. Diese werden nur benötigt wenn Sie selbst Programme erstellen wollen, die auf dem SvxLink-Projekt basieren

Hinweis: Für ALSA-basierte Systeme (wie Fedora >=2) wird zur Soundein- und Ausgabe die Alsa OSS-Emulation genutzt. Allerdings gibt es einen Fehler in der Emulationsschicht der bewirkt, dass SvxLink und Qtel nicht funktionieren. Um diesen Fehler zu umgehen wurde die Umgebungsvariable ASYNC_AUDIO_NOTRIGGER eingeführt.
Diese ist standardmäßig auf 1 gesetzt um diesen Fehler zum umgehen, da die meisten modernen Linuxdistributionen standardmäßig ALSA nutzen. Falls Sie mit der Audioausgabe Probleme bekommen sollten, setzen Sie diese Variable auf 0. Für den SvxLink-Server können Sie das im Startskript /etc/sysconfig/svxlink machen.
Für Qtel können Sie diese Variable manuell in der Kommandozeile, in einem Startskript oder im Loginskript der Shell setzen. Die nachfolgende Anleitung setzt als Shell die Bash voraus die bei den meisten Linuxdistributionen als Default-Shell eingestellt ist:

```
export ASYNC_AUDIO_NOTRIGGER=0
qtel &
```

Oder auf der Kommandozeile:

```
ASYNC_AUDIO_NOTRIGGER=0 qtel &
```

Die Umgebungseinstellungen gehen jedes Mal verloren, wenn Sie sich aus dem System ausloggen. Aus diesem Grund sollten Sie die oben aufgeführte Exportzeile in die Datei ".bash_profile" eintragen, welche Sie in Ihrem Home-Verzeichnis vorfinden. Das Setzen dieser Variable ist nicht zwangsläufig notwendig und kann dazu führen, dass SvxLink oder Qtel nicht ordnungsgemäß funktioniert. Setzen Sie deshalb diese Variable nur wenn Sie Audio-Probleme haben!

Hinweis: Beenden Sie alle anderen Applikationen, die auf die Soundkarte zugreifen und zur selben Zeit laufen wie SvxLink/Qtel. Wenn ein anderes Programm das Soundinterface bereits geöffnet hat, dann wartet SvxLink so lange bis diese Applikation beendet wird. Insbesondere im Falle von Problemen mit SvxLink, z.B. "Aufhängen" und weiteren Soundproblemen, prüfen Sie ob eventuell eins Soundserver wie z.B. artsd oder andere laufen. Wie dem auch sei, diese Problem ist nicht weiter wichtig, da die meisten Distributionen den ALSA-Sound-Layer benutzen.
Wenn Sie ausschließlich Qtel nutzen möchten, so lesen Sie zuerst den Abschnitt "Audio-Pegeleinstellungen" und im Anschluß die Qtel-Dokumentation.

Hardware
Um den SvxLink-Server zu betreiben ist eine bestimmte Hardware notwendig um den Computer mit dem Funkgerät zu verbinden. Im Moment nutze ich ein Interface von Aleks, S54S, welches den PC vollständig vom Funkgerät mittels Optokoppler und Transformatoren trennt.
Für lange Zeit habe ich ein einfaches, nicht isoliertes Interface betrieben welches auch funktioniert hatte. Allerdings habe ich auf diese Weise eine Soundkarte zerstört.

WB0RXX/Tim hat eine Schaltung entwickelt, welche er zusammen mit SvxLink nutzt. Diese Schaltung kann man sich hier anschauen.

Auch die für Echolink entwickelten Interfaces können für SvxLink genutzt werden. Schauen Sie einfach auf die Echolink-Webseite. Getestet habe ich keine von diesen, deshalb kann ich keine Garantie auf Funktion geben. Bitte geben Sie mir eine kurze Rückinformation, ob diese Interfaces zusammen mit SvxLink funktionieren oder nicht.

Audiopegel einstellen
Es gibt weder in Qtel noch im SvxLink-Server die Möglichkeit den Audiopegel einzustellen. Die Pegel müssen mit Hilfe von aumix, kmix, alsamixer oder einer anderen externen Applikation eingestellt werden.
Suchen Sie die Regler, die für die Einstellung der Pegel benötigt werden. Der NF-Ausgangspegel wird durch die Regler Pcm und Vol definiert und der Eingangspegel durch den Regler “Capture” (manchmal auch mit IGain bezeichnet), nicht aber der Mic bzw. line-in Regler! Die letzten beiden werden genutzt, um den Pegel für das Mithören der beiden Eingänge einzustellen.

Setzen Sie beide auf den Pegel 0. Wählen Sie entweder den Mic- oder den line-in Eingang aus.
Die Regler Pcm Vol und Mic/Input sollten bei etwqa 50% stehen. Stellen nun Sie die Pegel entsprechend der folggenden Anleitung ein.
Um die Pegel für Qtel einzustellen, verbinden Sie sich mit dem *ECHOTEST*-Server. Dieser EchoLinkserver sendet alles zurück, was er empfangen hat. Kurz nachdem die Verbindung aufgebaut wurde, hören Sie eine Begrüßung. Stellen Sie jetzt den Pegel mit Hilfe der Regler Pcm und Vol auf eine akzeptable Lautstärke ein. Betätigen Sie die PTT-Taste, sprechen Sie etwas und hören Sie dann wie das Gesprochene wiedergegeben wird. Verstellen Sie die Pegel des Capture-Reglers so lange bis Sie zufrieden sind.

Um die Audiopegel für den SvxLinkserver einzustellen, setzen Sie zunächst die Konfigurationsvariable PEAK_METER auf 1 im Abschnitt Empfängerkonfiguration der svxlin.conf.
Starten Sie den svxlink und drücken Sie *# auf Ihrer Tastatur. Dieses Kommando bewirkt, das sich SvxLink selbst identifiziert. Das Nummernzeichen # ist nur notwendig wenn das *-Kommando direkt auf der Tastatur eingegeben wird. Wenn Die dies per Funk machen möchten, so senden Sie ein * und das anschließende Schließen der Rauschsperre bewirkt den selben Effekt. Machen Sie das mehrere Male und stellen Sie die Regler für Pcm und Vol auf die höchstmöglichen Werte ohne dabei zu übersteuern.

Um den Audioeingangspegel einzustellen öffnen Sie die Rauschsperre des Empfängers, so dass SvxLink das Rauschen des Empfängers zugeführt wird.
Erhöhen Sie den Audioeingangspegel so lange, bis Meldungen über Übersteuerung auf der Konsole erscheine. Reduzieren Sie jetzt wieder den Pegel bis diese Meldungen nicht mehr ausgegeben werden. Falls diese Meldungen gar nicht erscheinen ist der Audiopegel insgesamt zu niedrig. In diesem Fall sollten Sie versuchen das Problem auf der analogen Eingangsseite zu lösen oder die PREAMP-Konfigurationsvariable im Abschnitt Empfängerkonfiguration entsprechend zu verändern.

Aktivieren Sie nun das Parrotmodul (Papagei) durch drücken von 1# auf ihrer Tastatur. Senden Sie jetzt mit einem anderen Funkgerät etwas und hören Sie, ob das wiedergegebene Signal eine gute Qualität besitzt. Senden Sie einige DTMF-Zeichen und schauen Sie, ob diese auch ordnungsgemäß dekodiert werden. Falls nicht, dann verändern Sie die Audiopegel erneut.

Testen Sie alle 16 Zeichen: 0-9, *, #, A, B, C, D.

Als Regel für alle Einstellungen gilt, setzen Sie niemals die Regler auf Werte über 90%. Bei den meisten Soundcards führt es zu Verzerrungen wenn der Pegel zu hoch ist. Falls der Audiopegel nicht ausreichend sein sollte so verändern Sie den Wert der Konfigurationsvariable PREAMP im Konfigurationsfile.

Konfiguration

Dies ist eigentlich der wichtigste Abschnitt, die Konfiguration von SvxLink. Von vielen Rückmeldungen der Nutzer wissen wir, dass hier teilweise erheblicher Klärungsbedarf besteht. Vielleicht können die folgenden Seiten einige Fragen klären und Ihnen eine Hilfestellung bei der Einrichtunges Ihres SvxLink-Nodes bieten. **Achten Sie bitte darauf, dass es für jedes einzelne Modul eine eigene Konfigurationsdatei gibt, diese Dateien befinden sich bei einer Standardinstallation unter***/etc/svxlink/svxlink.d*
Es ist also normalerweise nicht damit getan die *svxlink.conf* einzurichten, sondern Sie müssen danach in Abhängigkeit der von Ihnen konfigurierten Module auch noch diese einzeln einrichten!
Falls Fragen nicht beantwortet werden, so nutzen sie bitte das Forum, hier können spezifische Probleme oft viel schneller geklärt werden.
Verzeichnissstruktur

/etc/svxlink/ ...enthält die Konfigurationsdateien für SvxLink, den remotetrx sowie die SvxLink-Module

/etc/rc.d/

…die System-Startskripte *svxlink* und *remotetrx*

/usr/bin/

…hier sind die Binaries zu finden (*svxlink, qtel, remotetrx, siglevdetcal*)

/usr/include/svxlink

enthält die Async-Header-Dateien (z.B. *AsyncAudioSink.h*)

/usr/lib/

…die SvxLink-core-Bibliotheken wie z.B. *libasyncaudio-0.17.0.99rc1.so*

/usr/lib/svxlink

…die SvxLink-Modul-Bibliotheken wie z.B. *ModuleMetarInfo.so*

/usr/share/qtel

…die Translationfiles (Übersetzungen in 8 Sprachen) für Qtel

/usr/share/svxlink

…die *events.tcl*, das “Hauptskript” der Eventbehandlungsroutinen

/usr/share/svxlink/events.d

…hier liegen die logik- und modulspezifischen tcl-Files, die sozusagen die Präsentationsebene bilden, also Sprachausgaben generieren, Repeaterkennung als Morsezeichen ausgeben, etc.

/usr/share/svxlink/modules.d

…hier liegen weitere TCL’s für SvxLink-Module

/usr/share/svxlink/sounds

…die Soundfiles für die Sprachausgaben in weitere landes- bzw. sprachspezifische Unterverzeichnissen wie z.B. *de_DE/* oder *en_EN/*. Wird im Linuxsystem keine Ländereinstellung erkannt bzw. wurde es nicth explizit im Logik-Abschnitt definiert, erfolgt defaultmäßig die Suche in *en_US/*

/var/log

Hier liegt standardmäßig das SvxLink-Logfile *svxlink*, dass über Ereignisse und Fehler informiert. Wird SvxLink in der Konsole gestartet, erfolgen die Ausgabe nicht in diesem File sondern in der Konsole selbst.

/var/spool/svxlink

Spool-Verzeichnis für Mails, Sprachmailbox, Voice-Mitschnitte usw., wenn der QSO-Recorder aktiviert oder der PropagationMonitor konfiguriert wurde.

PropagationMonitor

Eine kurze Anleitung in englischer Sprache ist hier zu finden:
http://sourceforge.net/mailarchive/message.php?msg_id=23367461
1) Installation von postfix und procmail

Wenn diese beiden Programme noch nicht auf Ihrem Sstem installiert sein sollten, dann tun Sie das bitte.
Erstellen Sie den Nutzer *svxlink* sowie das Home-Verzeichnis */home/svxlink*
Erstellen Sie eine Datei *.procmailrc* und speichern Sie diese bitte im Home-Verzeichnis des Users svxlink ab (*/home/svxlink*). Hier ein Beispiel:

```
##############################################################
LOGFILE=/home/svxlink/procmail.log
VERBOSE=ON # debug, should be switched to OFF later
# Store received propagation alerts here
MAILDIR=/var/spool/svxlink/propagation_monitor
# Put e-mails from the DXrobot (gooddx.net) in the folder below
:0:
* ^From.*DXrobot
dxrobot
# Put e-mails from VHFDX (vhfdx.net) in the folder below
:0:
* ^From.*DX.Sherlock
vhfdx
# Send all other, non matching e-mails, to /dev/null.
# Replace "!root" with "/dev/null" to throw non-matching
# e-mails away instead or comment both lines below out
# to just let non-matching e-mails pass through.
:0:
/dev/null
#
# end of .procmailrc
##############################################################
```

2) Erstellen Sie die Mail-Spool-Verzeichnisse

/var/spool/svxlink/propagation_monitor
/var/spool/svxlink/propagation_monitor/vhfdx
/var/spool/svxlink/propagation_monitor/dxrobot

Der Eigentümer muß svxlink sein.

```
chown -R svxlink:users /var/spool/svxlink/propagation_monitor
```

3) Auf manchen Linux-Systemen muss postfix für die Annahme von eMails außerhalb des eigenen Netzwerks eingerichtet werden.
Editieren Sie hierfür das Konfigurationsfile *main.cf* AM ENDE der Datei(!):
/etc/postfix/main.cf

```
inet_interfaces = all
```

4) Prüfen Sie ob postfix beim Sytemstart mit gestartet wird, sehen Sie hierzu in der Datei*/etc/postfix/main.cf* nach

```
mailbox_command = /usr/bin/procmail
```

5) Auf Systemen mit geringen Resourcen sollte das “size limit” herabgesetzt werden um das System vor “vollaufenden” Fehlermeldungen zu schützen

```
queue_minfree = 0
mailbox_size_limit = 512000
```

message_size_limit = 128000
Auf unserem EPIA-Board mit 8GByte SD-Karte und 512MByte RAM + Ramdisk verweigert postfix ohne o.g. Änderungen die Annahme von eingehenden eMails. Wenn Sie genügend Resourcen haben, lassen Sie alles wie es ist.

6) Wenn Sie SvxLink an einem DSL-Anschluß mit täglich wechselnder IP-Adresse betreiben, ist die Anmeldung bei z.B. dyndns.org erforderlich. Zum Beispiel:*DEIN_RUFZEICHEN.homelinux.org*
Fügen Sie den Eintrag *DEIN_RUFZEICHEN.homelinux.org* in der Datei *main.cf* hinzu:
mydomain = DEIN_RUFZEICHEN.homelinux.org
Wenn Sie ein privates Netzwerk betreiben (ohne Hostnamen/Domains *.homelinux.org), dann muß postfix noch mitgeteilt werden, dass er für den Transport eingehender eMails an svxlink@DEIN_RUFZEICHEN.homelinux.org zuständig ist.

7) Richten Sie Ihr System so ein, dass postfix in den Runleveln 3 (und 5) gestartet wird

8) Richten Sie auf Ihrem DSL-Router ein Portforwarding für Port 25/tcp ein. Das Ziel muss auf die IP-Adresse des Rechners verweisen, auf dem Sie svxlink betreiben. Sie können die ordnungsgemäße Funktion testen, indem Sie
“telnet YOUR_CALLSIGN.homelinux.org 25″ von einem Rechner außerhalb Ihres eigenen Netzwerkes aus aufrufen.

9) Registrieren Sie sich in der vhfdx.net-Mailingliste mit Ihrer neuen eMail-Adresse*svxlink@DEIN_RUFZEICHEN.homelinux.org*
Nach einigen Minuten sollte die erste eMail eingehen, prüfen Sie ob in der Datei
/home/svxlink/procmail.log
ein entsprechender Eintrag erscheint.
Prüfen Sie weiterhin, dass diese eMail im richtigen Verzeichnis abgespeichert wurde:
/var/spool/svxlink/propagation_monitor/vhfdx
10) Installieren Sie alle benötigten Sounddateien für den Propagationmonitor.
/usr/share/svxlink/sounds/de_DE/PropagationMonitor
oder erstellen Sie eigene Dateien (z.B. mit Mary)

11) Fügen Sie in der Datei */etc/svxlink/svxlink.conf* den Eintrag zur Aktivierung des Propagation hinzu
[RepeaterLogic]
MODULES=...,ModulePropagationMonitor,...

oder in einer SimplexLogic
[SimplexLogic]
MODULES=...,ModulePropagationMonitor,...
und starten Sie SvxLink neu.

12) Der Propagationmonitor prüft jede Minute, ob eine neue eMail eingegangen ist. Diese eMail wird analysiert,aufbereitet, ausgegeben und danach in einen Archiv-Ordner verschoben. Prüfen Sie von Zeit zu Zeit, ob das Löschen alter eMails notwendig ist (Platzmangel)

LocationInfo (APRS)

SvxLink unterstützt sowohl das Senden von APRS-Baken zum APRS-Netzwerk als auch die Echolink-eigenen Statusmeldungen. Im APRS-Netzwerk wird dabei der Connect/Disconnect einer Station als APRS-Statusmessage angezeigt. Im APRS-Netzwerk wird dabei der SvxLink-Node als "Objekt von…" in der Form *EL-DL1ABC* bzw. *ER-DB0ABC* angezeigt und nicht als *EL-12345*. Es gibt verschiedene Gründe, weshalb dies in der vorliegenden Form implementiert wurde.
Die Konfiguration erfolgt in der Datei */etc/svxlink/svxlink.conf* durch das Eintragen der Variable LOCATION_INFO:

```
[GLOBAL]
...
LOCATION_INFO=LocationInfo
...

[LocationInfo]
APRS_SERVER_LIST=rotate.aprs.net:14580
STATUS_SERVER_LIST=aprs.echolink.org:5199
LON_POSITION=11.12.13E
LAT_POSITION=51.52.53N
FREQUENCY=438.725
CALLSIGN=ER-DB0ABC
TX_POWER=5
ANTENNA_GAIN=6
ANTENNA_HEIGHT=12m
ANTENNA_DIR=-1
PATH=WIDE1-1
BEACON_INTERVAL=10
TONE=0
COMMENT=SvxLink by SM0SVX (svxlink.sourceforge.net)
```

Beschreibung

LOCATION_INFO

Zeigt auf den Abschnitt innerhalb der */etc/svxlink/svxlink.conf*, die die Definitionen für die LocationInfo (APRS) enthält.

APRS_SERVER_LIST

Server des APRS-Netzwerkes, der die Positionsmeldungen/Statusinformationen entgegennimmt (via TCP). z.B *APRS_SERVER_LIST=rotate.aprs.net:14580*

STATUS_SERVER_LIST

Echolink-Server, der die aktuellen Positions- und Statusmeldungen entgegennimmt (via UDP), z.B. *STATUS_SERVER_LIST=aprs.echolink.org:5199*

LON_POSITION

Geographischer Längengrad der eigenen Position, z.B. *LON_POSITION=11.12.13E*

LAT_POSITION

Geographischer Breitengrad der eigenen Position, z.B. *LAT_POSITION=51.52.53N*

FREQUENCY

Sendefequenz des SvxLink-Nodes, z.B. *FREQUENCY=438.725*

CALLSIGN

Das Rufzeichen, unter dem SvxLink läuft, wobei ein "ER-" für Relaisstationen und "EL-" für Simplex-Links vorangestellt werden muß. z.B. *CALLSIGN=ER-DB0ABC*

TX_POWER

Die Sendeleistung in Watt, z.B. *TX_POWER=5*

ANTENNA_GAIN

Der Antennengewinn in dB, z.B. *ANTENNA_GAIN=6*

ANTENNA_HEIGHT

Die Antennenhöhe über Grund, NICHT MSL(!), in Meter (*ANTENNA_HEIGHT=12m*) oder Fuss (*ANTENNA_HEIGHT=22*).

ANTENNA_DIR

Die Hauptstrahlungsrichtung der Sendeantenne in Grad, beim Rundstrahler "-1″ eintragen,
z.B. *ANTENNA_DIR=-1*

PATH

Der APRS-Pfad, nur wichtig, wenn die Bake von einem benachbarten APRS-Digi wieder ausgesendet wird,
z.B. *PATH=WIDE1-1*

BEACON_INTERVAL

Das Sende-Intervall in Minuten, z.B. *BEACON_INTERVAL=10*. Kleinere Werte als 10 werden automatisch auf 10 Minuten gesetzt.

TONE

Gibt an, ob für den Funkverkehr über das Relais ein Rufton oder CTCSS-Ton benötig wird, z.B.
TONE=0 -> kein Rufton/CTCSS notwendig
TONE=1750 -> Rufton 1750 Hz zum Öffnen des Relais
TONE=250 -> CTCSS-Ton von 250Hz notwendig

COMMENT

Kommentar, der bei jeder Bake mit gesendet wird, z.B.
COMMENT=[Svx] Relais Musterstadt

RepeaterLogic

Um einen Repeater einzurichten, müssen Sie in der Datei */etc/svxlink/svxlink.conf* den Abschnitt [RepeaterLogic] bearbeiten und diesen Abschnitt im [GLOBAL]-Bereich aktivieren. Der Name in der Konfigurationsvariable LOGICS=RepeaterLogic muß dabei exakt mit der Abschnittsbezeichnung [RepeaterLogic] übereinstimmen, also auch in Groß- und Kleinschreibung.

Datei /etc/svxlink/svxlink.conf

```
[GLOBAL]
LOGICS=...,RepeaterLogic,...

[RepeaterLogic]
TYPE=Repeater
RX=Rx
TX=Tx
MODULES=ModuleHelp,ModuleParrot,ModuleEchoLink,ModuleMetarInfo
CALLSIGN=DL1ABC-L
SHORT_IDENT_INTERVAL=60
LONG_IDENT_INTERVAL=120
IDENT_ONLY_AFTER_TX=4
#EXEC_CMD_ON_SQL_CLOSE=500
EVENT_HANDLER=/usr/share/svxlink/events.tcl
RGR_SOUND_DELAY=250
#REPORT_CTCSS=136.5
#TX_CTCSS=ALWAYS
MACROS=Macros
LINKS=LinkToR4
FX_GAIN_NORMAL=0
FX_GAIN_LOW=-12
ACTIVATE_MODULE_ON_LONG_CMD=4:EchoLink
MUTE_RX_ON_TX=1
#NO_REPEAT=1
SQL_FLAP_SUP_MIN_TIME=500
SQL_FLAP_SUP_MAX_COUNT=7
OPEN_ON_1750=0
OPEN_ON_SQL=500
CLOSE_ON_SEL5=12345
OPEN_SQL_FLANK=CLOSE
#OPEN_ON_CTCSS=250.6:20
QSO_RECORDER=7:QsoRecorder
```

<u>Beschreibung</u>

TYPE

Definiert den Logictyp, also Simplex-, Repeater- (oder Phonelogik), z.B. TYPE=Simplex

RX

Verweis auf den Abschnitt innerhalb der */etc/svxlink/svxlink.conf*, in dem der Empfänger (Rx) konfiguriert wird, z.B. RX=Rx

TX

Verweis auf den Abschnitt innerhalb der */etc/svxlink/svxlink.conf*, in dem der Sender (Tx) konfiguriert wird, z.B. TX=Tx

MODULES

Kommaseparierte Liste von Modulen, die innerhalb dieser Logik aktiviert und zur Verfügung stehen sollen, z.B. MODULE=ModuleHelp,ModuleParrot,ModuleEchoLink,ModuleMetarInfo

CALLSIGN

Rufzeichen, unter dem dieser SimplexLink arbeiten soll, z.B. CALLSIGN=DL1ABC-L

SHORT_IDENT_INTERVAL

Interval in Minuten, nachdem eine kurze Identifizierung erfolgt. Default: Ausgabe des Rufzeichens. Konfiguration in der SimplexLogic.tcl.
z.B. SHORT_IDENT_INTERVAL=60

LONG_IDENT_INTERVAL

Interval in Minuten, nachdem eine lange Identifizierung erfolgt. Default: Ausgabe des Rufzeichens und der Uhrzeit. Konfiguration in der SimplexLogic.tcl. z.B. LONG_IDENT_INTERVAL=120

IDENT_ONLY_AFTER_TX

Dieser Parameter steuert wann die Identifikation ausgeführt wird. Standardmäßig wird die Identifikation jedes Mal ausgeführt, wenn die im Parameter*SHORT_IDENT_INTERVAL* konfigurierte Zeit (in Minuten) abgelaufen ist. Wird der Parameter aktiviert, so erfolgt die Identifizierung nur, wenn gerade gesendet wird. Gedacht ist dieses Feature für den Fall, dass der Node ein Repeater über HF ansteuert und die Identifizierung dieses Links nicht notwendig ist bzw. stören würde.
z.B. IDENT_ONLY_AFTER_TX=1

EXEC_CMD_ON_SQL_CLOSE

Zeit in Millisekunden nach dem Schließen der Rauschsperre, innerhalb dieser ein Kommando auch ohne abschließende Raute (#) als Kommando erkannt wird. Dieser Parameter ist gedacht für Funkgeräte, bei denen das Senden von DTMF-Zeichen relativ schwierig ist (VX-2). Das Einschalten dieses Parameters kann ungewollte Effekte hervorrufen, z.B. dass mitten im QSO ein DTMF-Zeichen erkannt und automatisch ein Modul gestartet wird.

EVENT_HANDLER

Die Position des für diese Logik zuständige Eventhandler-File, z.B.
EVENT_HANDLER=/usr/share/svxlink/events.tcl

RGR_SOUND_DELAY

Verzögerungszeit in Millisekunden nach dem Schließen der Squelch um einen Roger-Piep auszulösen, z.B. RGR_SOUND_DELAY=250

REPORT_CTCSS=136.5

Wenn konfiguriert, erfolgt die Ansage der konfigurierten CTCSS-Frequenz sobald man mit * die manuelle Identifizierung auslöst.
z.B. REPORT_CTCSS=136.5. Wenn der RX keinen CTCSS-Ton benötigt sollte dieser Parameter auskommentiert werden.

TX_CTCSS

Konfiguriert in einer RepeaterLogic, unter welchen Umständen ein CTCSS-Ton mit ausgesendet wird: ALWAYS oder SQL_OPEN. Im ersten Fall wird der CTCSS-Ton zugeschaltet, sobald die Station auf Sendung geht. Im zweiten Fall erst wenn eine Audioausgabe erfolgt, welche per HF empfangen wurde. Dieser Parameter wird nur bei Relaisstationen berücksichtigt.
Mögliche Parameter: ALWAYS,SQL_OPEN,LOGIC,MODULE,ANNOUNCEMENT

MACROS

Namen des Abschnittes innerhalb der */etc/svxlink/svxlink.conf*, die die Macrodefinitionen enthält
z.B. MACROS=Macros Es wird dann in der svxlink.conf ein Abschnitt erwartet der mit*[Macros]* beginnt

LINKS

Namen des Abschnittes innerhalb der */etc/svxlink/svxlink.conf*, der die Link-Definitionen enthält, also die Kopplung mit anderen Logiken auf dem selben SvxLink-Node ermöglicht, z.B. LINKS=Link_to_R2

FX_GAIN_NORMAL

Lautstärke der Ansagen und Sounds wenn auf dem Relais kein weiterer Funkverkehr druchgeführt wird. In diesem Falle werden alle Ansagen um den konfigurierten Betrag (in dB) abgesenkt (oder auch verstärkt).
z.B. FX_GAIN_NORMAL=0

FX_GAIN_LOW

Lautstärke der Ansagen und Sounds wenn auf dem Relais Funkverkehr druchgeführt wird. In diesem Falle werden alle Ansagen um den konfigurierten Betrag (in dB) abgesenkt.
z.B. FX_GAIN_LOW=-12

ACTIVATE_MODULE_ON_LONG_CMD
Automatische Aktivierung eines konfigurierten Modules, wenn ein langes Kommando mit der konfigurierten Anzahl von Digits empfangen wird. D.h., das Modul muß vorher nicht erst explizit aktiviert werden.
z.B. ACTIVATE_MODULE_ON_LONG_CMD=4:EchoLink -> startet das Modul EchoLink, wenn ein mindestens 4stelliges Kommando empfangen wurde
MUTE_RX_ON_TX
Wenn =1, dann wird das RX-Audio-Signal während der Sendung stumm geschalten. Wenn nicht erforderlich sollte dieser Parameter auskommentiert werden.
NO_REPEAT
Wenn =1, dann gibt SvxLink NICHT das empfangene Audiosignal an den TX weiter. Nutzen sie diese Einstellung, wenn Sie einen Repeater betreiben, bei dem die NF-Zweige von TX und RX direkt gekoppelt sind. Der Nachteil ist in diesem Fall, dass SvxLink die empfangene RX-NF nicht weiterverarbeiten kann (Clipper, Deemphasis, DTMF-Töne ausblenden, etc.). Wird die NF innerhalb von SvxLink verarbeitet, so entsteht systembedingt eine geringe Verzögerung, die für den praktischen Funkbetrieb aber keine große Rolle spielt.
IDLE_TIMEOUT
Anzahl der Sekunden bis der Sender nach dem Auftasten wieder abfällt (Abfallzeit).
z.B. IDLE_TIMEOUT=10
OPEN_ON_1750
Zeit in Millisekunden, die ein Rufton von 1750Hz empfangen werden muß bis der Repeater öffnet. Ein Wert von 0 deaktiviert das Öffnen auf einen Rufton.
OPEN_ON_CTCSS
Wenn dieser Parameter konfiguriert wurde, öffnet der Repeater auf den entsprechenden CTCSS-Ton (PL). Der Eingabesyntax ist ton_frequenz:min_länge. Die Tonfrequenz wird in Hertz angegeben und die Tonlänge in Millisekunden. z.B. für einen CTCSS-Ton von 136.5Hz und einer gewünschten Tonlänge von 1 Sekunde muß*OPEN_ON_CTCSS=136:1000* eingetragen werden.
OPEN_ON_DTMF
Bei diesem Parameter kann festgelegt werden, ob ein einzelnes DTMF-Digit den Repeater öffnen kann. z.B. OPEN_ON_DTMF=*
OPEN_ON_SQL
Bei diesem Parameter wird die Zeit (in Millisekunden) einer geöffneten Squelch als Kriterium für das Öffnen des Repeaters herangezogen.
z.B. OPEN_ON_SQL=500 -> Die Rauschsperre muß mindestens 0.5 Sekunden geöffnet bleiben, damit der Repeater öffnet
OPEN_ON_SEL5
Öffnet den Repeater wenn der entsprechende Mehrtonruf ordnungsgemäß empfangen wurde, z.B. OPEN_ON_SEL5=02311
CLOSE_ON_SEL5
Schließt den Repeater wenn der entsprechende Mehrtonruf ordnungsgemäß empfangen wurde, z.B. CLOSE_ON_SEL5=02123
OPEN_SQL_FLANK
Hier kann festgelegt werden, ob der Repeater auf eine steigende oder fallende Flanke öffnet. Voraussetzung ist, dass entweder OPEN_ON_SQL oder OPEN_ON_CTCSS als Öffnungs-Kriterium konfiguriert wurden.
Bei *OPEN_SQL_FLANK=OPEN* öffnet der Repeater und startet die Übertragung des Audiostreams augenblicklich. Es wird keine Identifizierung ausgegeben.
Bei *OPEN_SQL_FLANK=CLOSE* öffnet der Repeater erst, wenn die Squelch wieder schließt. In diesem Fall wird eine Identifizierung ausgesendet.
IDLE_SOUND_INTERVAL
Wenn der Repeater geöffnet ist, aber kein Funkverkehr darüber abgewickelt wird, dann kann mit der Konfiguration dieses Parameters die Ausgabe eines Sounds erreicht werden. Der Wert entspricht dabei der Zeit zwischen diesen “Erinnerungstönen”.
z.B. IDLE_SOUND_INTERVAL=1000
SQL_FLAP_SUP_MIN_TIME

Dieser Parameter ermöglich das Schließen des Repeaters im Fall von Interferenzen (flatternde Rauschsperre). Der Wert ist die Zeit in Millisekunden, die ein Signal mindestens anliegen muß um als reales Rx-Signal erkannt zu werden. Gute Werte bewegen sich im Bereich von 500ms – 1000ms

SQL_FLAP_SUP_MAX_COUNT

Dieser Wert gibt die maximale Anzahl von kurzen Squelchöffnungen (sog. Bursts) an, die empfangen werden dürfen. Wird dieser Wert überschritten, so wird der Sender des Repeaters abgeschaltet. Eingünstiger Wert ist 5 bis 10.

SEL5_MACRO_RANGE

Definiert einen Bereich für eingehende Selektivrufe, die als Macro erkannt werden sollen.
z.B. *SEL5_MACRO_RANGE=01200, 01399* Wird bei dieser Konfiguration eine 5Ton-Folge “01225″ empfangen, so geht SvxLink davon aus, dass im [Macro]-Bereich ein entsprechender Eintrag konfiguriert wurde und versucht diesen auszuführen. Andere 5Ton-Folgen, wie z.B. “23312″ werden von SvxLink ignoriert.

DEFAULT_LANG

Definiert die Sprachumgebung und ist wichtig für das Finden der jeweiligen Sprachumgebung. Wird der Eintrag nicht angegeben, so wird die Srachumgebung des Linuxsystemes genutzt.
z.B. DEFAULT_LANG=de_DE

QSO_RECORDER

Kommando und Verweis auf den Konfigurationsabschnitt in dem der QSO-Recorder konfiguriert wird. Die Nummer vor dem Doppelpunkt definiert das Aktivierungs-/Deaktivierungskommando. Die Bezeichnung danach die Bezeichnung des Konfigurationsabschnitts.
Bsp: OSQ_RECORDER=8:QsoRecorder

Mit 81# wird der QQSO-Recorder gestartet und mit 80# gestoppt.

SimplexLogic

Um einen einfachen Link einzurichten, müssen Sie in der Datei */etc/svxlink/svxlink.conf*den Abschnitt [SimplexLogic] bearbeiten und diesen Abschnitt im [GLOBAL]-Bereich aktivieren. Der Name in der Konfigurationsvariable *LOGICS=SimplexLogic* muß dabei exakt mit der Abschnittsbezeichnung *[SimplexLogic]* übereinstimmen, also auch in Groß- und Kleinschreibung.

Datei /etc/svxlink/svxlink.conf

```
[GLOBAL]
LOGICS=...,SimplexLogic,...

[SimplexLogic]
TYPE=Simplex
RX=Rx
TX=Tx
MODULES=ModuleHelp,ModuleParrot,ModuleEchoLink,ModuleMetarInfo
CALLSIGN=DL1ABC-L
SHORT_IDENT_INTERVAL=60
LONG_IDENT_INTERVAL=120
IDENT_ONLY_AFTER_TX=4
#EXEC_CMD_ON_SQL_CLOSE=500
EVENT_HANDLER=/usr/share/svxlink/events.tcl
RGR_SOUND_DELAY=250
#REPORT_CTCSS=136.5
#TX_CTCSS=ALWAYS
MACROS=Macros
LINKS=LinkToR4
FX_GAIN_NORMAL=0
FX_GAIN_LOW=-12
ACTIVATE_MODULE_ON_LONG_CMD=4:EchoLink
ACTIVATE_MODULE_ON_STARTUP=EchoLink
MUTE_RX_ON_TX=1
#OSQ_RECORDER=8:QsoRecorder
```

Beschreibung

TYPE

Definiert den Logictyp, also Simplex-, Repeater- oder Phonelogik
z.B. TYPE=Simplex

RX

Verweis auf den Abschnitt innerhalb der */etc/svxlink/svxlink.conf*, in dem der Empfänger (Rx) konfiguriert wird, z.B. RX=Rx

TX

Verweis auf den Abschnitt innerhalb der */etc/svxlink/svxlink.conf*, in dem der Sender (Tx) konfiguriert wird, z.B. TX=Tx

MODULES

Kommaseparierte Liste von Modulen, die innerhalb dieser Logik aktiviert und zur Verfügung stehen sollen, z.B. MODULE=ModuleHelp,ModuleParrot,ModuleEchoLink,ModuleMetarInfo

CALLSIGN

Rufzeichen, unter dem dieser SimplexLink arbeiten soll, z.B. CALLSIGN=DL1ABC-L

SHORT_IDENT_INTERVAL

Interval in Minuten, nachdem eine kurze Identifizierung erfolgt. Default: Ausgabe des Rufzeichens. Konfiguration in der SimplexLogic.tcl.
z.B. SHORT_IDENT_INTERVAL=60

LONG_IDENT_INTERVAL

Interval in Minuten, nachdem eine lange Identifizierung erfolgt. Default: Ausgabe des Rufzeichens und der Uhrzeit. Konfiguration in der SimplexLogic.tcl. z.B. LONG_IDENT_INTERVAL=120

IDENT_ONLY_AFTER_TX

Dieser Parameter steuert wann die Identifikation ausgeführt wird. Standardmäßig wird die Identifikation jedes Mal ausgeführt, wenn die im Parameter*SHORT_IDENT_INTERVAL* konfigurierte Zeit (in Minuten) abgelaufen ist. Wird der Parameter aktiviert, so erfolgt die Identifizierung nur, wenn gerade gesendet wird. Gedacht ist dieses Feature für den Fall, dass der Node ein Repeater über HF ansteuert und die Identifizierung dieses Links nicht notwendig ist bzw. stören würde.
z.B. IDENT_ONLY_AFTER_TX=1

EXEC_CMD_ON_SQL_CLOSE

Zeit in Millisekunden nach dem Schließen der Rauschsperre, innerhalb dieser ein Kommando auch ohne abschließende Raute (#) als Kommando erkannt wird. Dieser Parameter ist gedacht für Funkgeräte, bei denen das Senden von DTMF-Zeichen relativ schwierig ist (VX-2). Das Einschalten dieses Parameters kann ungewollte Effekte hervorrufen, z.B. dass mitten im QSO ein DTMF-Zeichen erkannt und automatisch ein Modul gestartet wird.

EVENT_HANDLER

Die Position des für diese Logik zuständige Eventhandler-File, z.B.
EVENT_HANDLER=/usr/share/svxlink/events.tcl

RGR_SOUND_DELAY

Verzögerungszeit in Millisekunden nach dem Schließen der Squelch um einen Roger-Piep auszulösen, z.B.
RGR_SOUND_DELAY=250

REPORT_CTCSS=136.5

Wenn konfiguriert, erfolgt die Ansage der konfigurierten CTCSS-Frequenz sobald man mit * die manuelle Identifizierung auslöst.
z.B. REPORT_CTCSS=136.5. Wenn der RX keinen CTCSS-Ton benötigt sollte dieser Parameter auskommentiert werden.

TX_CTCSS

Konfiguriert in einer RepeaterLogic, unter welchen Umständen ein CTCSS-Ton mit ausgesendet wird: ALWAYS oder SQL_OPEN. Im ersten Fall wird der CTCSS-Ton zugeschaltet, sobald die Station auf Sendung geht. Im zweiten Fall erst wenn eine Audioausgabe erfolgt. Dieser Parameter wird nur bei Relaisstationen berücksichtigt.
z.B. TX_CTCSS=ALWAYS

MACROS

Namen des Abschnittes innerhalb der */etc/svxlink/svxlink.conf*, die die Macrodefinitionen enthält
z.B. MACROS=Macros Es wird dann in der svxlink.conf ein Abschnitt erwartet der mit*[Macros]* beginnt

LINKS

Namen des Abschnittes innerhalb der */etc/svxlink/svxlink.conf*, der die Link-Definitionen enthält, also die Kopplung mit anderen Logiken auf dem selben SvxLink-Node ermöglicht, z.B. LINKS=Link_to_R2

FX_GAIN_NORMAL

Lautstärke der Ansagen und Sounds wenn auf dem Relais kein weiterer Funkverkehr durchgeführt wird. In diesem Falle werden alle Ansagen um den konfigurierten Betrag (in dB) abgesenkt (oder auch verstärkt).
z.B. FX_GAIN_NORMAL=0

FX_GAIN_LOW

Lautstärke der Ansagen und Sounds wenn auf dem Relais Funkverkehr druchgeführt wird. In diesem Falle werden alle Ansagen um den konfigurierten Betrag (in dB) abgesenkt.
z.B. FX_GAIN_LOW=-12

ACTIVATE_MODULE_ON_LONG_CMD

Automatische Aktivierung eines konfigurierten Modules, wenn ein langes Kommando mit der konfigurierten Anzahl von Digits empfangen wird. D.h., das Modul muß vorher nicht erst explizit aktiviert werden.
z.B. ACTIVATE_MODULE_ON_LONG_CMD=4:EchoLink -> startet das Modul EchoLink, wenn ein mindestens 4stelliges Kommando empfangen wurde

MUTE_RX_ON_TX

Wenn =1, dann wird das RX-Audio-Signal während der Sendung stumm geschalten. Wenn nicht erforderlich

sollte dieser Parameter auskommentiert werden.
SEL5_MACRO_RANGE
Definiert einen Bereich für eingehende Selektivrufe, die als Macro erkannt werden sollen.
z.B. SEL5_MACRO_RANGE=01200, 01399 Wird bei dieser Konfiguration eine 5Ton-Folge "01225″ empfangen, so geht SvxLink davon aus, dass im [Macro]-Bereich ein entsprechender Eintrag konfiguriert wurde und versucht diesen auszuführen. Andere 5Ton-Folgen, wie z.B. "23312″ werden von SvxLink ignoriert.

DEFAULT_LANG
Definiert die Sprachumgebung und ist wichtig für das Finden der jeweiligen Sprachumgebung. Wird der Eintrag nicht angegeben, so wird die Srachumgebung des Linuxsystemes genutzt.
z.B. DEFAULT_LANG=de_DE

VOICELOGGER_DIR
Verzeichnis, in welches die Mitschnitte des Funkverkahrs auf dem Repeater gespeichert werden sollen. z.B. VOICELOGGER_DIR=/var/spool/svxlink/voice

VOICELOGGER_CMD
Kommando, mit welchem der Mitschnitt getartet bzw. gestopt werden kann, z.B. VOICELOGGER_CMD=8 Im praktischen Betrieb startet das Kommando 81# den Mitschnitt und das Kommando 80# stoppt die Aufzeichnung wieder.

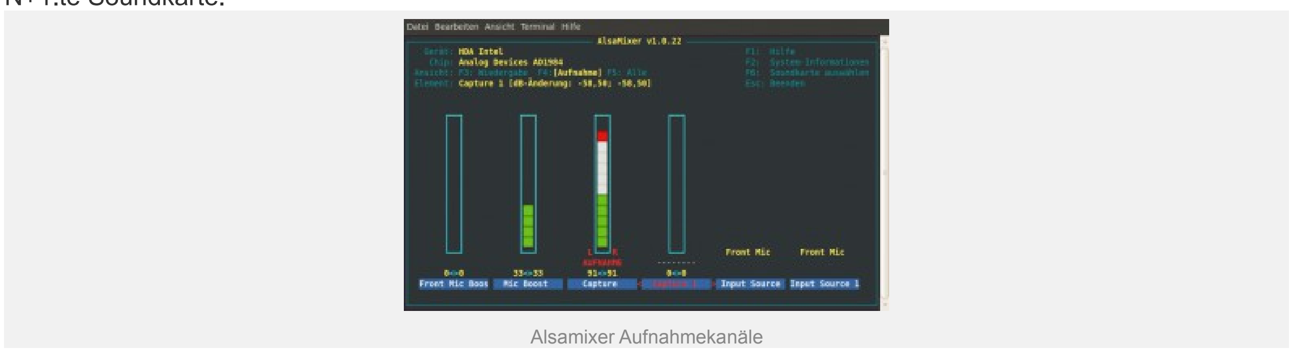
QSO_RECORDER
Kommando und Verweis auf den Konfigurationsabschnitt in dem der QSO-Recorder konfiguriert wird. Die Nummer vor dem Doppelpunkt definiert das Aktivierungs-/Deaktivierungskommando. Die Bezeichnung danach die Bezeichnung des Konfigurationsabschnitts.
Bsp: OSQ_RECORDER=8:QsoRecorder

Mit 81# wird der QQSO-Recorder gestartet und mit 80# gestoppt.

Audioeinstellungen

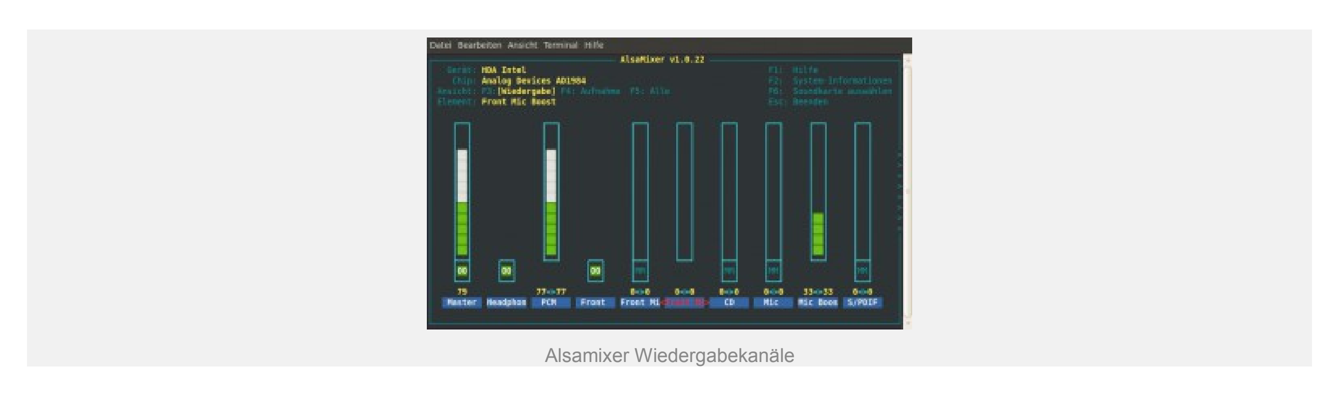
Zur Laufzeit müssen die Audiopegel der Soundkarte an die örtlichen Gegebenheiten angepaßt werden. Hier gibt es viele Varianten/Programme, stellvertretend soll die Variante mit dem *alsamixer* erläutert werden.
1) Loggen Sie sich auf dem SvxLink-Rechner lokal oder per ssh ein und starten sie SvxLink

2) Starten sie den *alsamixer*, ohne Option wird die erste Soundkarte angesprochen, sollte SvxLink die zweite, dritte, n-te Soundkarte nutzen, so starten Sie den alsamixer wie folgt:*alsamixer -cX* wobei X die Nummer der Soundkarte angibt, die SvxLink nutzt. Die Zählung beginnt bei 0 für die Erste, 1 für die Zweite, usw. .. N für die N+1.te Soundkarte.



Alsamixer Aufnahmekanäle

Die Bedienung erfolgt mit den Pfeiltasten, sowie den Funktionstasten [F1..F6]



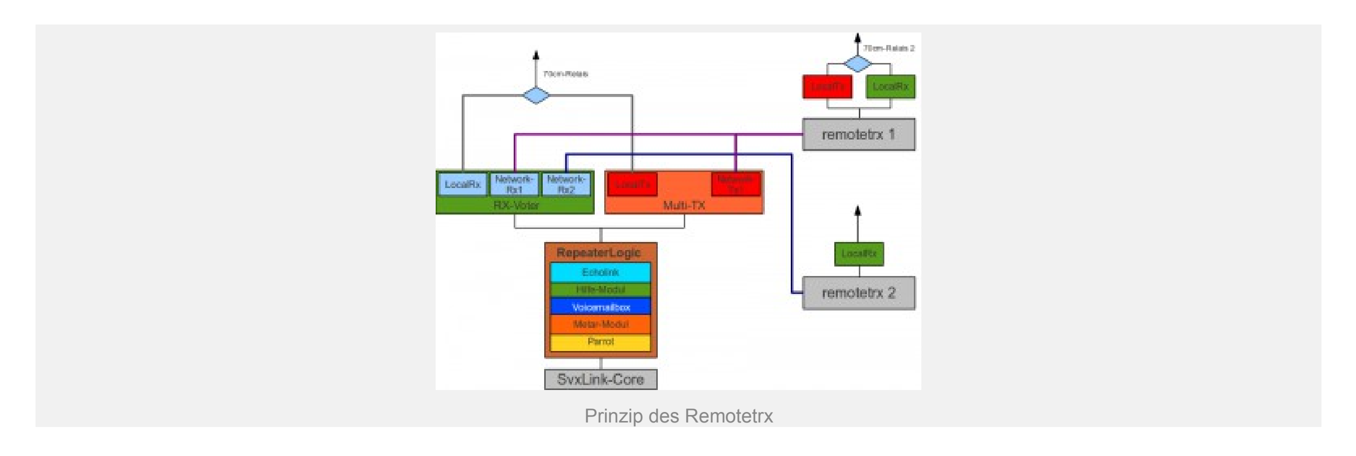
Alsamixer Wiedergabekanäle

3) Verändern Sie die Einstellungen entsprechend, deaktivieren Sie alle nicht benötigten Kanäle (Stummschalten per Taste [M]). I.A. benötigt man nur die Kanäle *Master*, *PCM*,*Capture* und e.v. *Mic Boost*
4) Die Regler sollten nicht auf 100% stehen

5) Verlassen Sie den Alsamixer uns speichern Sie die Einstellungen mit “*alsactl store X*” wobei X die Nummer der Soundkarte angibt (siehe oben)

remotetrx

Ein Remotetrx kann für den Betrieb von abgesetzten Sendern oder Empfängern genutzt werden. Die Einsatzfälle sind recht vielfältig. SvxLink bietet eine Voterfunktion, bei der die von verschiedenen Empfängern eingehenden Signale analysiert und das jeweils “beste” Empfangssignal zum Sender/den Sendern durchgeschaltet wird. Das folgende Bild soll eine Anwendungsmöglichkeit aufzeigen.

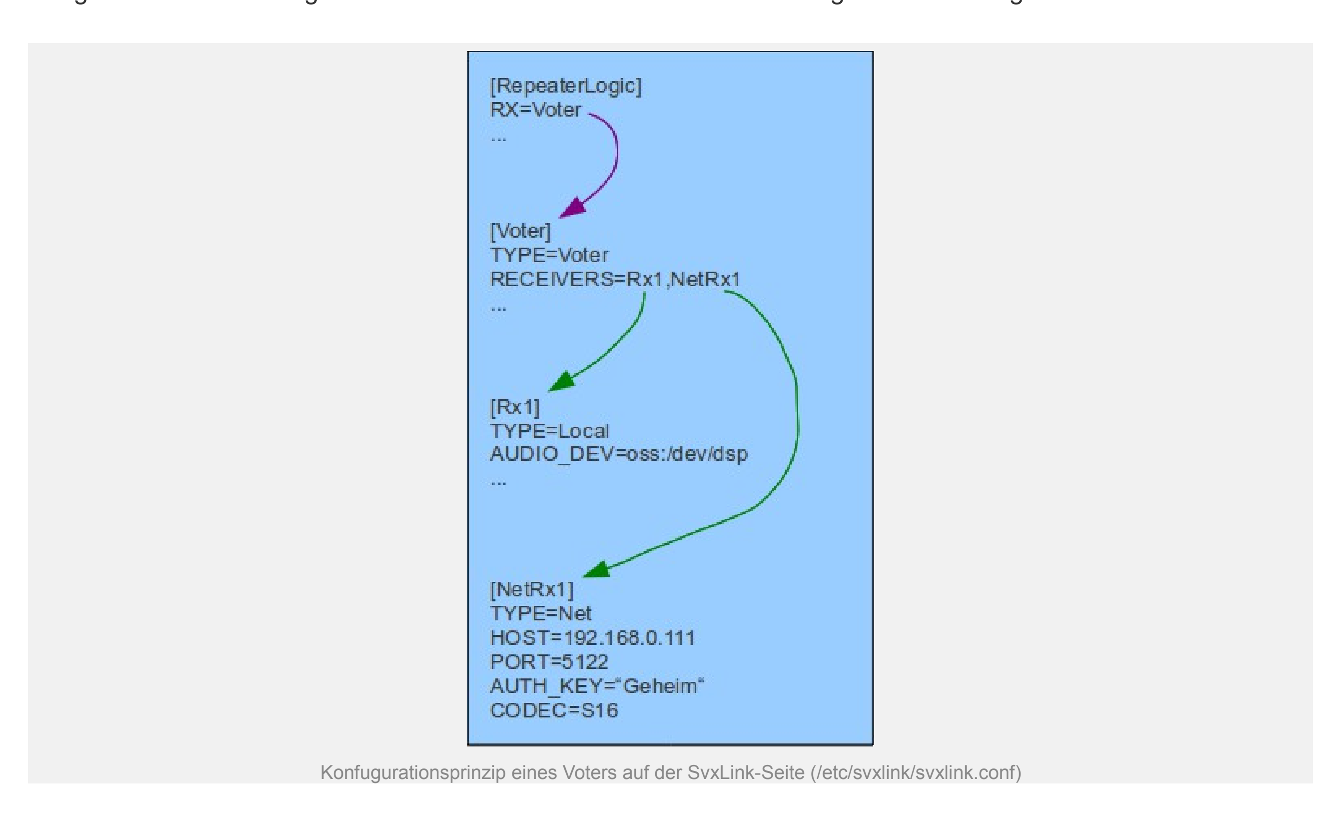


Prinzip des Remotetrx

Die Konfiguration auf der SvxLink-Base erfolgt in der */etc/svxlink/svxlink.conf*. Auf der remotetrx-Seite erfolgt die Konfiguration in der Datei */etc/svxlink/remotetrx.conf*.

1) SvxLink-Base

Die grundsätzliche Konfiguration des RX-Pfades für den Voter ist im folgenden Bild dargestellt



Konfugurationsprinzip eines Voters auf der SvxLink-Seite (/etc/svxlink/svxlink.conf)

/etc/svxlink/svxlink.conf

[RepeaterLogic]

...

RX=Voter

TX=Multi

...

[Voter]

TYPE=Voter

RECEIVERS=Rx1,NetRx1

VOTING_DELAY=200

[Rx1]

...

hier ist ein LocalRx konfiguriert, siehe Abschnitt "RX-Konfiguration"

...

Abschnitt für den RemoteRx

[NetRx1]

TYPE=Net

HOST=192.168.0.99

TCP_PORT=5213

AUTH_KEY="sehr_geheim"

CODEC=S16

Beschreibung

TYPE

Gibt an ob es sich um eine lokale (=Local) oder remote-Komponente (=Net) handelt.

HOST

Die IP-Adresse oder Hostname auf dem der Remotetrx läuft.

TCP_PORT

Der Port, über den die TCP-Kommunikation erfolgt

AUTH_KEY

Das Passwort, mit dem sich remotetrx und SvxLink-Basis gegenseitig authentifizieren. Groß- und Kleinschreibung ist wichtig.

CODEC

Der zu verwendende Sprachcodec, mit dem die Audiodaten zwischen SvxLink und remotetrx übertragen werden. Es stehen folgende Codecs zur Verfügung:

RAW, S16, SPEEX, GSM

Macros

Macros sind Aktionen auf dem Node, die beim Empfang eines bestimmten Kommandos ausgelöst werden.
Allgemein werden bei SvxLink Macros durch Voranstellen des DTMF-Digit's "D" eingeleitet, z.B. *D1234#*
Konfiguriert werden Macros in einem eigenen Abschnitt, der durch einen Abschnittsnamen gekennzeichnet wird und auf den von einem Logik-Konfigurationsabschnitt (in der */etc/svxlink/svxlink.conf*) aus verwiesen werden muß.
Aktivierung der Macros, durch einen logikspezifischen Eintrag in der*/etc/svxlink/svxlink.conf*

[RepeaterLogic]

...

MACROS=Macro_repeater

...

In der */etc/svxlink/svxlink.conf* muß jetzt ein Abschnitt *[Macro_repeater]* vorhanden sein, in dem die jeweiligen Macros definiert werden.

[Macro_repeater]

0=SelCallEnc:0112345#

1=EchoLink:9999#

2=MetarInfo:12#

...

Im ersten Beispiel wird beim Empfang des Macros "D0#" ein Selektivruf im Modus "01″ (=ZVEI-1) und der Nummer "12345″ ausgelöst.
Im zweiten Beispiel erfolgt beim Empfang des Macros "D1#" der Aufbau einer Echolink-Verbindung mit dem EchoLink-Testserver.
Das dritte Beispiel startet beim Empfang von "D2#" das Modul MetarInfo und die Ausgabe der Wettermeldungen der 12. konfigurierten Wetterstation.

Voraussetzung für alles ist natürlich, dass die jeweiligen Module entsprechend konfiguriert wurden.

Link(ing)

Es geht hierbei um die Möglichkeit, auf Anforderung zwei oder mehr Logiken miteinander zu verbinden. Interessant ist diese Anwendung, wenn z.B. an einem Standort sowohl ein 2m als auch 70cm-Repeater betrieben werden. Diese können bei Bedarf, z.B. während eines Rundspruchs miteinander “verlinkt” werden. In Kürze wird die Möglichkeiten bestehen, die Art und Weise freier definieren zu können, z.B. die automatische und permanente Verbindung beim Start von SvxLink usw.

Bitte dieses “Linking” nicht mit den Simplex-Links verwechseln.

Aktivierung durch einen Eintrag im jeweiligen Bereich einer Logik innerhalb der*/etc/svxlink/svxlink.conf*:

```
[RepeaterLogic_70cm]
...
LINKS=TestLink
...
[RepeaterLogic_2m]
...
LINKS=TestLink
...
[TestLink]
NAME=2Relais
CONNECT_LOGICS=RepeaterLogic_70cm:32:70cm,RepeaterLogic_2m:33:2m
DEFAULT_ACTIVE=1
AUTOACTIVATE_ON_SQL=1
TIMEOUT=300
```

Beschreibung

CONNECT_LOGICS

Hier werden die Logiken und Befehle konfiguriert, die miteinander verbunden werden sollen. Das Format ist Logicname1:Befehl1:Bezeichnung1,Logikname2:Befehl2:Bezeichnung1

Beispiel:

CONNECT_LOGICS=DB0ABC:34:ABC,DB0XYZ:72:XYZ

Um beide Logiken miteinander zu verbinden muß von DB0ABC aus das Kommando 341# aufgerufen werden, von DB0XYZ ist es das Kommando 721#.

Die kommandos 340# sowie 720# führen von der jeweiligen Logik aus zur Trennung der Verbindung.

DEFAULT_ACTIVE

Wenn dieser Parameter =1 ist, werden beide Logiken sofort nach dem Start von SvxLink verbunden.

TIMEOUT

Zeit in Sekunden, nach der die verbundenen Logiken wieder getrennt werden, falls kein Funkverkehr darüber geführt wurden.

NAME

Bezeichnung für diese Konfiguration, wird beim Verbinden den TCL-Skripten übergeben und kann z.B. beim Verbinden/Trennen mit angesagt werden

Qso-Recorder

Mit Hilfe des QSO-Recorders kann man ein laufendes QSO, welches gerade auf dem Relais/Link stattfindet aufzeichnen.

Aktivierung innerhalb eines Logic-Abschnittes durch, z.B.:

[SimplexLogic]

...

QSO_RECORDER=8:QsoRecorder

Konfiguration:

[QsoRecorder]

REC_DIR=/var/spool/svxlink/qso_recorder

MAX_TIME=3600

MIN_TIME=200

SOFT_TIME=300

MAX_DIRSIZE=1024

DEFAULT_ACTIVE=1

TIMEMOUT=100

QSO_TIMEOUT=300

ENCODER_CMD=

Beschreibung:

REC_DIR

Verzeichnis in dem die aufgezeichneten QSO’s als WAV-Datei abgelegt werden sollen.
Empfehlung */var/spool/svxlink/qso_recorder*

MIN_TIME

Wenn die Länge des aufgezeichneten Audiostreams kürzer ist als per MIN_TIME konfiguriert, dann wird das File sofort wieder gelöscht, nachdem die Aufzeichnung beendet ist. Standardeinstellung=0, leere Dateien werden gelöscht.

MAX_TIME

Hier wird die maximale Aufzeichnungszeit pro Audiofile definiert. Wird die hier definierte Aufzeichnungslänge überschritten, so wird das aktuelle File geschlossen und ein neues File göffnet. Standardeinstellung=0 (unbegrenzt)

SOFT_TIME

Um keine abrupten Unterbrechungen in der Aufzeichnung zu erhalten ist es möglich diese SOFT_TIME zu konfigurieren. Beispiel: Als MAX_TIME wurden 3600 Sekunden (1 Stunde) konfiguriert. Wenn SOFT_TIME=300 konfiguriert wurde, so wird versucht in der Zeit von 55 Minuten bis 60 Minuten ein Schließen der Rauschsperre auszunutzen um die Aufzeichnung zu beenden und gleich mit einem neuen Audiofile fortzuführen. Hierbei wird erricht, dass während einer Audioübertragung der Stream in zwei Stücke zerteilt wird.

MAX_DIRSIZE

Definiert die maximale Größe in MegaByte als Summe aller Files im Verzeichnis. Wird diese Größe überschritten, so werden die ältesten Files gelöscht. Die Verzeichnisgröße wird bei jedem Schließen eines Audiofiles überprüft. Hierbei werden nur Files mit dem Präfix “qso_rec_” berücksichtigt. Dies sollte beachtet werden, wenn man ENCODING_CMD verwendet. Das dort konfigurierte Kommando sollte den Präfix “qso_rec” nicht entfernen, wenn sichergestellt sein soll, dass die Verzeichnisgröße nicht überschritten werden soll. Standardeinstellung=0 (keine Begrenzung)

DEFAULT_ACTIVE

Wenn diese Variable =1 ist, dann wird der QSO-Recoder automatisch bei jedem SvxLink-Start aktiviert. Standardeinstellung=0 (inaktiv)

QSO_TIMEOUT

Wenn diese Variable gesetzt ist, wird das momentan geöffnete File nach jedem QSO geschlossen. Definiert wird die Anzahl in Sekunden, die ein Repeater/Link nicht genutzt wird um ein QSO als beendet zu erkennen. Standardeinstellung=0 (kein QSO Timeout)

ENCODER_CMD
Hier wird ein Befehl (Shell-Befehl) definiert, der jedes Mal ausgeführt wird, wenn die Aufzeichnung eines Files beendet wurde. Er wurde eingerichtet um beispielsweise eine Konvertierung des aufgezeichnten Files von *.wav nach *.mp3 anzustoßen. Allerdings können auch umfangreichere Aktionen ausgeführt werden wie z.B. das Kopieren der Files auf einen anderen Computer.
Dieses Kommando läuft unter der Shell, so dass die normalen Shell-Operatoren (Umleitung, Pipes, …) verwendet werden können

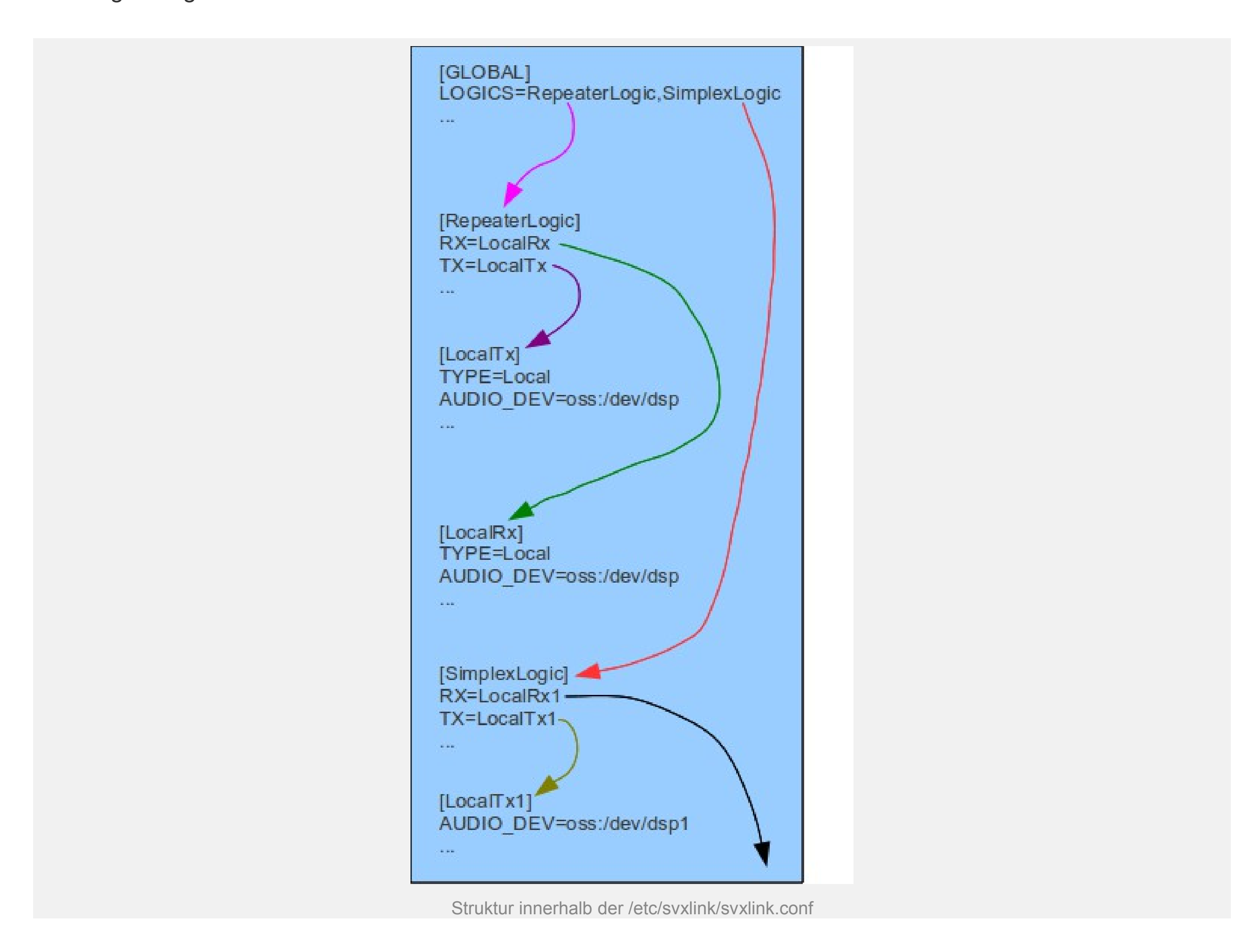
Routereinstellungen

Für die Einstellungen Ihres DSL-Routers gelten grundsätzlich die selben Parameter wie beim Betrieb der herkömmlichen Echolink-Software.
Für den Port 5200/tcp muß der Aufbau einer ausgehenden Verbindung erlaubt sein, die Ports 5199/udp und 5198/udp benötigen einen Portforward auf dem Router zu dem Rechner, auf dem SvxLink läuft.
Soll von außen der Zugriff per Qtel auf Ihren SvxLink-Server möglich sein, so muss ebenfalls für den Port 5200/tcp ein Portforward eingestellt werden.

Achten Sie auch auf eventuelle Firewalls, die auf Ihrem Rechner laufen, bei OpenSuSE ist z.B. standardmäßig ein Firewall aktiviert, der je nach Einstellung und Konfiguration die Funktionsweise von SvxLink einschränkt.

svxlink.conf

Die Datei */etc/svxlink/svxlink.conf* ist die Hauptkonfigurationsdatei von SvxLink. In Ihr wird festgelegt welche Logik(en) betrieben werden soll(en) (Repeater und/oder Simplex), welchen Rauschsperrentyp man für den jeweiligen Rx nutzen will, ob es sich um eine lokale Station handelt oder ob man abgesetzte Stationen (sogenannte remotetrx's) mit anbinden möchte usw.
Je nach eigener Zielstellung und vorhandener Hardware, die angebunden werden soll, kann diese Datei eine nicht ganz unerhebliche Größe erreichen, so dass man u.U. ein wenig die Übersicht verlieren kann. Das nächste Bild zeigt den grundsätzlichen strukturellen Aufbau dieser Datei.



Struktur innerhalb der /etc/svxlink/svxlink.conf

Die Namen innerhalb der eckigen Klammern ([Name]) kennzeichnen dabei Abschnitte, die sozusagen Teil- bzw. Unterkonfigurationen enthalten. Mit Ausnahme von *[GLOBAL]*sind diese Namen frei wählbar. Was in diesem Abschnitt konfiguriert wird, muss mit dem Parameter *TYPE=xxx* definiert werden.
Achten Sie aber nach Möglichkeit auf die Verwendung von sprechenden Bezeichnungen, wir empfehlen für eine "SimplexLogic" auch einen entsprechenden Namen zu verwenden, sonst kommen Sie früher oder später durcheinander! Achten Sie weiterhin auf eine exakte Übereinstimmung der Bezeichnungen insbesondere auf die Groß- und Kleinschreibung.
Die Konfigurationsabschnitte der /etc/svxlink/svxlink.conf

[GLOBAL]

MODULE_PATH=/usr/lib/svxlink

LOGICS=RepeaterLogic,AnalogPhone,GSMLogic,MicSpkrLogic1,MicSpkrLogic

CFG_DIR=/etc/svxlink.d

TIMESTAMP_FORMAT="%c"

CARD_SAMPLE_RATE=16000

LOCATION_INFO=LocationInfo

#LINKS=TestLink

Der [GLOBAL]-Abschnitt enthält grundsätzliche Definitionen, wie die Art und Anzahl der zu aktivierenden Logiken.

MODULE_PATH

Der Pfad, in dem die Modul-Bibliotheken zu finden sind.

LOGICS

Die Namen der Logiken, welche man gerne nutzen möchte.

CFG_DIR

Das Verzeichnis, in welchem die Konfigurationsdateien für die Module zu finden sind (z.B. ModuleMetarInfo.conf).

CARD_SAMPLE_RATE

Die Samplingrate [Hz], mit der auf der Soundkarte gearbeitet wird (8000, 16000, 22050, 44100, 48000). Einfache Soundkarten unterstützen oft nur 8000 und 48000.

TIMESTAMP_FORMAT

Format des Zeitstempels, der im Log erscheint, für detailierte Informationen siehe C++ Formatdeskriptoren.

LOCATION_INFO

Aktiviert den Bereich für die APRS-Positions- und Statusmeldungen, die SvxLink periodisch generiert.

LINKS

Kommaseparierte Liste von Abschnittsbezeichnungen, die Konfigurationsparameter zum Verbinden von verschiedenen Logiken enthalten.

Logic-Abschnitt

RX-Adapter

Der RX-Adapter realisiert den Empfangsteil der SvxLink-Applikation. Er ist für den Empfangssignalweg und damit verbunden Funktionen zuständig. Hierzu gehören:

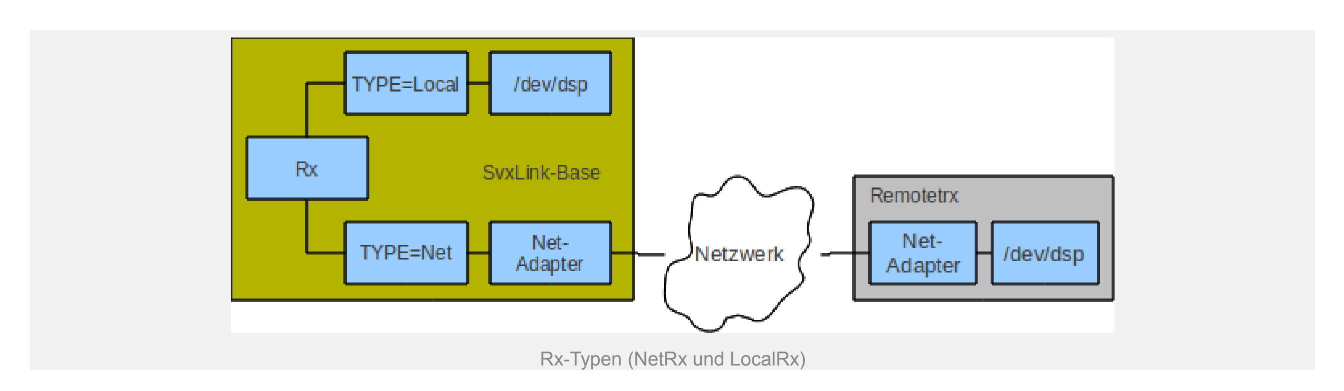
- Aufbereitung und Verarbeitung des Empfangssignales (NF-seitig)
- Generierung eines Squelch-Kriteriums
- Dekodierung von Tonfrequenzen wie DTMF, Sel5, CTCSS und normalen Ruftönen

Beispiel:

```
[Rx1]
TYPE=Local
AUDIO_DEV=alsa:plughw:0
AUDIO_CHANNEL=0
SQL_DET=VOX
SQL_START_DELAY=10
SQL_DELAY=10
SQL_HANGTIME=3000
#SQL_TIMEOUT=600
VOX_FILTER_DEPTH=20
VOX_THRESH=1000
CTCSS_FQ=136.5
CTCSS_THRESH=10
SERIAL_PORT=/dev/ttyS0
SERIAL_PIN=CTS:SET
#EVDEV_DEVNAME=/dev/input/by-id/usb-SYNIC_SYNIC_Wireless_Audio-event-if03
#EVDEV_OPEN=1,163,1
#EVDEV_CLOSE=1,163,0
#SIGLEV_DET=TONE
SIGLEV_SLOPE=1
SIGLEV_OFFSET=0
#TONE_SIGLEV_MAP=100,84,60,50,37,32,28,23,19,8
```

SIGLEV_OPEN_THRESH=30
SIGLEV_CLOSE_THRESH=10
DEEMPHASIS=0
#SQL_TAIL_ELIM=300
#PREAMP=6
PEAK_METER=0
DTMF_DEC_TYPE=INTERNAL
DTMF_MUTING=1
DTMF_HANGTIME=100
DTMF_SERIAL=/dev/ttyS0
#DTMF_MAX_FWD_TWIST=8
#DTMF_MAX_REV_TWIST=4
#1750_MUTING=1
#SEL5_DEC_TYPE=INTERNAL
#SEL5_TYPE=ZVEI1

Es stehen grundsätzlich zwei Varianten zur Verfügung: ein LocalRx, der die auf dem Server vorhandenen Audiointerfaces direkt anspricht (/dev/dsp bzw. hwplug:0) sowie ein NetRx, der den Empfangspfad eines über das Netzwerk angebundenen Servers (remotetrx) ansprechen kann. Das folgende Bild stellt diese Möglichkeiten prinzipiell dar.



Rx-Typen (NetRx und LocalRx)

TYPE

Definiert ob es sich um einen lokalen oder einen per Netzwerk angeschlossenen Rx handelt, z.B. *TYPE=Local*

AUDIO_DEV

Das zu verwendende Audiodevice *AUDIO_DEV=oss:/dev/dsp* oder*AUDIO_DEV=alsa:plughw:0*

AUDIO_CHANNEL

Die Nummer desStereo-Kanales, der verwendet wird. Im Gegensatz zum herkömmlichen Echolink lassen sich beide Stereokanäle getrennt verwenden, wenn es die Soundkarte unterstützt!
z.B. *AUDIO_CHANNEL=0* oder *AUDIO_CHANNEL=1*

SQL_DET

SQL_DET=CTCSS -> Rauschsperrentyp ist “CTCSS”, das heißt, nur wenn ein CTCSS-Ton empfangen wird, gilt es als Öffnen der Rauschsperre
SQL_DET=SIGLEV -> Signal-Detector, der den am Eingang anliegenden Rauschpegel auswertet. Die Rauschsperre des RX muß offen sein und es muß eine Signalpegelkalibrierung durchgeführt werden (/usr/bin/siglevdetcal)
SQL_DET=VOX -> Rauschsperrentyp ist VOX, also praktisch die NF des angeschlossenen Funkgerätes
SQL_DET=SERIAL -> Rauschsperrentyp ist Serial, d.h. es wird ein über die serielle Schnittstelle eingehendes (externes) Signal genutzt.
SQL_DET=EVDEV -> Wertet den Status des Interfaces /dev/input/eventX aus, z.B. Taste auf einer USB-Soundkarte

SQL_START_DELAY

Zeit in ms, die vergehen muß, damit das konfigurierte Rauschsperrenkriterium das Ereignis als Öffnen der Rauschsperre erkennt. Hier sind Werte >0 einzutragen, falls eine VOX als Rauschsperre verwendet wird,

z.B. *SQL_START_DELAY=100*.

SQL_DELAY

Zeit in ms, um das das Ereignis “SQL_OPEN” verzögert wird (Weitergabe an das SvxLink-System), Einsatzfälle sind z.B beim SigLevdetector oder beim RX-Voting. Wertebereich 20-100

SQL_HANGTIME

Zeit in ms, die die Rauschsperre noch offen bleibt, auch nachdem der Rauschsperrendetector ein Schließen registriert hat, z.B. *SQL_HANGTIME=75*

EVDEV_DEVNAME

Konfiguriert das Geräte /dev/input welches für den EVEDEV-Squelchdetektor genutz wird.

EVDEV_OPEN

Nutzen Sie das Programm *evtest* um den Typ, Code und Wert herauszufinden umd die Rauschsperre zu öffnen, Bsp: erhalten Sie die Paramter typ=1,code=163,value=1 so ist der Parameter EVDEV_OPEN wie folgt zu konfigurieren: *EVDEV_OPEN=1,163,1*

EVDEV_CLOSE

Nutzen Sie das Programm *evtest* um den Typ, Code und Wert herauszufinden umd die Rauschsperre zu schließen, Bsp: erhalten Sie die Paramter typ=1,code=163,value=0 so ist der Parameter EVDEV_CLOSE wie folgt zu konfigurieren: *EVDEV_CLOSE=1,163,0*

CTCSS_FQ

Wenn *SQL_DET=CTCSS*, dann wird die hier konfigurierte CTCSS-Frequenz als Rauschsperrenkriterium verwendet., z.B. *CTCSS_FQ=136.5*

CTCSS_THRESH

nicht mehr genutzt

CTCSS_OPEN_THRESH

Wenn als Rauschsperrentyp CTCSS konfiguriert wurde (*SQL_DET=CTCSS*), dann definiert dieser Parameter den Pegel für das Öffnen der Rauschsperre. Dieser Pegel stellt dabei eine Art Signal/Nutzsignal-Verhältnis [dB] dar. Wenn Sie CTCSS-Modus 2 oder 3 nutzen, dann ist es hilfreich den Parameter CTCSS_SNR_OFFSET zu konfigurieren. Der Standardwert für CTCSS_OPEN_THRESH ist 15dB.

CTCSS_CLOSE_THRESH

Wenn als Rauschsperrentyp CTCSS konfiguriert wurde (*SQL_DET=CTCSS*), dann definiert dieser Parameter den Pegel für das Schließen der Rauschsperre. Dieser Pegel stellt dabei eine Art Signal/Nutzsignal-Verhältnis [dB] dar. Wenn Sie CTCSS-Modus 2 oder 3 nutzen, dann ist es hilfreich den Parameter CTCSS_SNR_OFFSET zu konfigurieren. Der Standardwert für CTCSS_CLOSE_THRESH ist 9dB.

CTCSS_SNR_OFFSET

Dieser Parameter wird nur berücksichtigt, wenn Sie den Parameter CTCSS_MODE auf 0, 2 oder 3 gesetzt haben. Er kalibriert des ermittelten SNR Wert so, dass er einem realen SNR nahekommt. Dieser Parameter muß bei jeder Änderung der Werte CTCSS_FQ, CTCSS_MODE, CTCSS_BPF_LOW und CTCSS_BPF_HIGH neu ermittelt werden.

Es besteht keine Notwendigkeit, diesen Parameter zu setzen. In diesem Fall müssen Sie aber mehr mit den Parametern CTCSS_OPEN_THRESH und CTCSS_CLOSE_THRESH experimentieren um korrekte Einstellungen für die Rauschsperre zu finden.

CTCSS_MODE

Diese Variable definiert die CTCSS-Erkennungsmethode. Folgende Werte können stehen zur Verfügung:

- 0 (Default) – Diese Methode ist die Standardeinstellung. Momentan ist es die “Estimated SNR”-Methode
- 1 (Neighbour bins) – Diese Methode nutzt drei schmale Frequenzbänder (~8Hz) zur Erkennung. Ein Band liegt im Zentrum des zu erkennenden Tones, die anderen darüber bzw. darunter. Diese Frequenzbänder werden genutzt um den Rauschpegel zu berechnen. In SvxLink wurde diese Methode lange Zeit verwendet, allerdings ist sie sehr langsam (~450ms) und sollte nicht mehr angewendet werden. Allerdings bleibt sie für den Fall verfügbar, dass bei bestimmten Hardware-Konfigurationen die anderen Methoden u.U. nicht verwendet werden können.
- 2 (Estimated SNR) – Dies ist eine neuere Implementierung mit einigen Verbesserungen, z.B. einer wesentlich schnelleren Detektierungszeit (~200ms). Es ist das Standardverfahren, wenn nicht konfiguriert wurde. Dieses Verfahren nutzt eine höhere Bandbreite um das Grundrauschen zu bestimmen und ist damit stabiler. Standardmäßig wird das gesamte CTCSS-Band ausgewertet, kann aber mit den Parametern CTCSS_CPF_LOW

und CTCSS_BPF_HIGH angepasst werden.

- 3 (Estimated SNR+Phase) – Dieser Detektor ist noch im Experimentierstadium. Er ist etwas schneller und wesentlich schmalbandiger als die anderen Verfahren und die Detektierungszeit liegt bei etwa 150ms. Damit ist er unempfindlicher gegenüber angrenzenden Tönen allerdings auch unempfindlicher beim zu detektierenden Ton.

SERIAL_PORT
Wenn *SQL_DET=SERIAL*, dann ist hier der serielle Port einzustellen, über die der Rauschsperrenstatus vom externen Rx geliefert wird, z.B. *SERIAL_PORT=/dev/ttyUSB2*

SERIAL_PIN
Wenn *SQL_DET=SERIAL*, dann ist hier der Pin und Pegel zu definieren, an dem das externe Signal anliegt, z.B. *SERIAL_PIN=CTS:SET* -> wenn an CTS ein high-Pegel anliegt gilt die Rauschsperre als offen

SERIAL_SET_PINS
Setzt die Pins einer seriellen Schnittstelle auf einen definierten Wert (H oder L). Das kann beispielsweise genutzt werden um eine Referenzspannung zum Pegel-Vergleich zu erzeugen. Hierbei können nur zwei Pins verwendet werden: RTS oder DTR.
Beispiel: SERIAL_SET_PINS=RTS!DTR setzt RTS auf H- und DTR auf L-Pegel

TONE_SIGLEV_MAP
Diese Variable weist Töne Signalpegeln zu wenn DIGLEV_DET=TONE. Es ist eine kommaseparierte Liste von Werten im Bereich 0-100. Der erste Wert korreliert mit der Frequenz 5500Hz, der Zweite mit 5600Hz … der letzte Wert mit 6400Hz. Welche Werte konfiguriert werden hängt von der Implementierung ab. Die Standardeinstellung ist 10,20,30,40,…100.

SIGLEV_DET
Wählen Sie hier den Typ des Signaldetektors. Es gibt zwei Auswahlmöglichkeiten: NOISE oder TONE. Der SIGLEV_DET wird nur benötigt, wenn Sie mehrere Empfänger in einer Voter-Konfiguration betreiben oder als Rauschsperrentyp SIGLEV konfiguriert wurde.
Der Detektortyp NOISE nutzt ein Bandpassfilter im Bereich 5-5.5kHz (bei CARDSAMPLE_RATE >= 16000) oder einen Hochpass (bei CARDSAMPLE_RATE = 8000) um den Betrag des Rauschanteils zu bestimmen.
Der Detektortyp TONE ist kein wirklicher Signalpegeldetektor sondern mehr ein Mechanismus um einen Pegel von einem Remote-TRX zu erhalten, der per HF angekoppelt ist.

SIGLEV_OPEN_THRESH
Wenn *SQL_DET=SIGLEV*, dann gibt dieser Wert den Pegel [%] an bei dem die Rauschsperre öffnet

SIGLEV_CLOSE_THRESH
Wenn *SQL_DET=SIGLEV*, dann gibt dieser Wert den Pegel [%] an bei dem die Rauschsperre schließt

SIGLEV_SLOPE
Wenn *SQL_DET=SIGLEV*, dann hier die "Verstärkung" des SigLev-Detectors eintragen, siehe Rauschsperrenkalibrierung.

SIGLEV_OFFSET
Wenn *SQL_DET=SIGLEV*, dann hier den "Offset" des SigLev-Detectors eintragen, siehe Rauschsperrenkalibrierung.

DEEMPHASIS
Höhenabsenkung des RX-Signales um den hier konfigurierten Betrag in [dB]

SQL_TAIL_ELIM
Dieser Parameter kann genutzt werden, um das Rauschen nach Beendigung einer Übertragung bis zum Einsetzen der Rauschsperre zu minimieren. Die empfangenen Audiodaten werden um den im Parameter *SQL_TAIL_ELIM* definierten Betrag in ms gepuffert. Beim Schließen der Rauschsperre werden die verbleibenden Audiopakete (einsetzendes Rauschen) nicht weiter übertragen. Achtung: Der gesamt über den Sender übertragene RX-Audiostream wird um diesen Betrag verzögert.

PREAMP
Das eingehende RX-Signal wird um den hier konfigurierten Betrag verstärkt [dB].

PEAK_METER
Gibt eine Warnmeldung im */var/log/svxlink* aus, falls der anliegende RX-Audiopegel zu hoch sein sollte. (=1 Warnungen aktiviert, =0 Warnungen deaktiviert)

DTMF_DEC_TYPE

Gibt den verwendeten DTMF-Decodertyp an, z.B. *DTMF_DEC_TYPE=INTERNAL*verwendet den in SvxLink eingebauten DTMF-Softwaredecoder, DTMF_DEC_TYPE=S54erwartet einen an die serielle Schnittstelle angeschlossenen DTMF-Hardwaredecoder.

DTMF_MUTING

Wenn =1, dann werden empfangene DTMF-Töne nicht über den Sender ausgegeben, nur wenn RX und TX audioseitig nicht direkt gekoppelt sind und *NO_REPEAT* nicht =1 ist.

DTMF_HANGTIME

Ist die Zeit in ms, die ein DTMF-Signal mindestens empfangen werden muß, damit es sicher als solches erkannt werden kann. Empfohlener Wertebereich 50-100ms

DTMF_SERIAL

Serielle Schnittstelle, an die ein S54-DTMF-Dekoder angeschlossen ist, z.B.*DTMF_SERIAL=/dev/ttyS1*

DTMF_MAX_FWD_TWIST

DTMF besteht aus zwei verschiedenen Tonfrequenzen (niedrige und höhere), die zeitgleich gesendet werden und normalerweise die gleiche Amplitude haben sollten. Das ist praktisch nie der Fall. *DTMF_MAX_FWD_TWIST* gibt an, um wieviel dB niedriger die Amplitude des höheren Tones gegenüber dem niedrigeren Ton sein darf. z.B.*DTMF_MAX_FWD_TWIST=8*

DTMF_MAX_REV_TWIST

Wie *DTMF_MAX_FWD_TWIST*, nur umgekehrt (wenn der hohe Ton eine um*DTMF_MAX_REV_TWIST* höhere Amplitude heben darf.)

1750_MUTING

Wenn =1, dann wird bei Empfang des 1750Hz-Ruftones der Sender stummgetastet. Die Funktion ist nur möglich, wenn RX und TX audioseitig nicht direkt gekoppelt sind und*NO_REPEAT* nicht =1 ist.

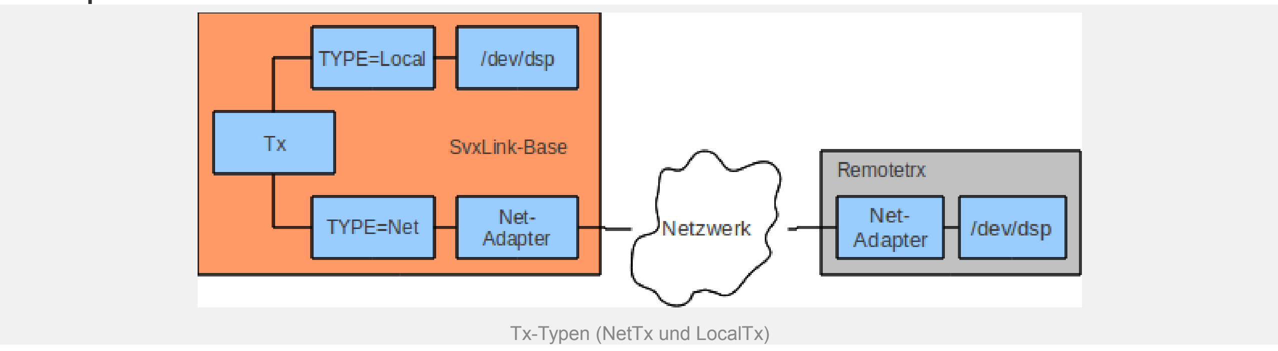
SEL5_DEC_TYPE=INTERNAL

Gibt an, ob ein interner oder externer Mehrtondekoder genutzt werden soll, momentan darf hier nur INTERNAL konfiguriert werden

SEL5_TYPE

Definiert den zu dekodierenden Mehrton-Standard (ZVEI1,EEA,EIA,CCITT,CCIR,…), z.B.*SEL5_TYPE=ZVEI1*

TX-Adapter



Tx-Typen (NetTx und LocalTx)

Beispiel:

[Tx1]

TYPE=Local

AUDIO_DEV=alsa:plughw:0

AUDIO_CHANNEL=0

PTT_PORT=NONE

PTT_PIN=DTRRTS

#PTT_HANGTIME=1000

TIMEOUT=300

TX_DELAY=500

#CTCSS_FQ=136.5

#CTCSS_LEVEL=9

PREEMPHASIS=0

DTMF_TONE_LENGTH=100

DTMF_TONE_SPACING=50

DTMF_TONE_AMP=-18

TYPE

Gibt den Typ an, also lokaler Sender oder über Netzwerk angebunden, z.B. *TYPE=Local*

AUDIO_DEV

Das Audiogerät (Lautsprecherausgang) der Soundkarte, der als TX genutz wird. Beim OpenSound-System ist vor der Gerätedefinition ein “oss” voranzustellen, z.B.*AUDIO_DEV=oss:/dev/dsp1* bei ALSA ein “alsa”, z.B. *AUDIO_DEV=alsa:plughw:1*

AUDIO_CHANNEL

Der Kanal des Ausgangs der genutzt wird. Im Gegensatz zum herkömmlichen Echolink, können vei SvxLink die Stereo-Kanäle getrennt verwendet werden, z.B.*AUDIO_CHANNEL=1*

PTT_PORT

Der Port, der für die RX/TX-Umschaltung genutzt wird, z.B. *PTT_PORT=/dev/ttyS0*

PTT_PIN

Der Pin des PTT-Ports, der den Pegel für die RX/TX-Umschaltung liefert, z.B.

PTT_PIN=DTRRTS -> bei TX sind DTR und RTS auf high

PTT_PIN=DTR ->bei TX ist nur DTR high

PTT_PIN=!DTR -> Invertierung, falls der angeschlossene Sender einen low-Pegel benötigt um auf Sendung zu gehen

PTT_PIN=NONE -> keine PTT erforderlich, z.B. bei der MicSpkrLogic

TIMEOUT

Maximale Sendezeitbegrenzung in Sekunden, z.B. *TIMEOUT=3000*. Der TX schaltet nach einer Sendezeit von 3000 Sekunden unabhängig von anderen Kriterien ab.

TX_DELAY

Auftastverzögerung des Senders in ms, z.B. *TX_DELAY=0*

CTCSS_FQ

Der auf dem TX-Träger aufmodulierte CTCSS-Ton, z.B. *CTCSS_FQ=136.5*

CTCSS_LEVEL

Der Audiopegel des CTCSS-Tones, z.B. *CTCSS_LEVEL=9*

PREEMPHASIS

Höhenanhebung des Audiosignales um x dB

Links

Im Link-Abschnitt wird konfiguriert, welche Logiken auf Anforderung zusammengeschalter werden. Im Moment wird das automatisierte “Linken” der Logiken noch nicht unterstützt, so dass dies per Hand durch ein Kommando passieren muß. Folgende Variablen können konfiguriert werden: *CONNECT_LOGICS, NAME, COMMAND, TIMEOUT*

CONNECT_LOGICS enthält eine kommaseparierte Liste der zu linkenden Logiken, z.B.*CONNECT_LOGICS=RepeaterLogic,SimplexLogic,MicSpkrLogic*.

NAME ist eine freie Definition, wird beim Linken ausgegeben, z.B. NAME=R2Link.

TIMEOUT wird momentan nicht genutzt.

COMMAND Das Kommando um die Logiken miteinander zu verbinden oder zu trennen, z.B.*COMMAND=94* mit dem Kommando 941# werden die konfigurierten Logiken verbunden und mit 940# wieder getrennt

Macros

Auf diesen Macro-Abschnitt wird von den einzelnen Logiken heraus mit Hilfe der Variablen *MACROS*= aus verwiesen. Prinzipiell kann für jeder Logik ein eingener Macro-Bereich zugeordnet werden, so dass jede Logik verschiedene Macos beinhalten könnte. Weitere Informationen zur Konfiguration von Makros sind unter http://svxlink.de/?page_id=458 zu finden.

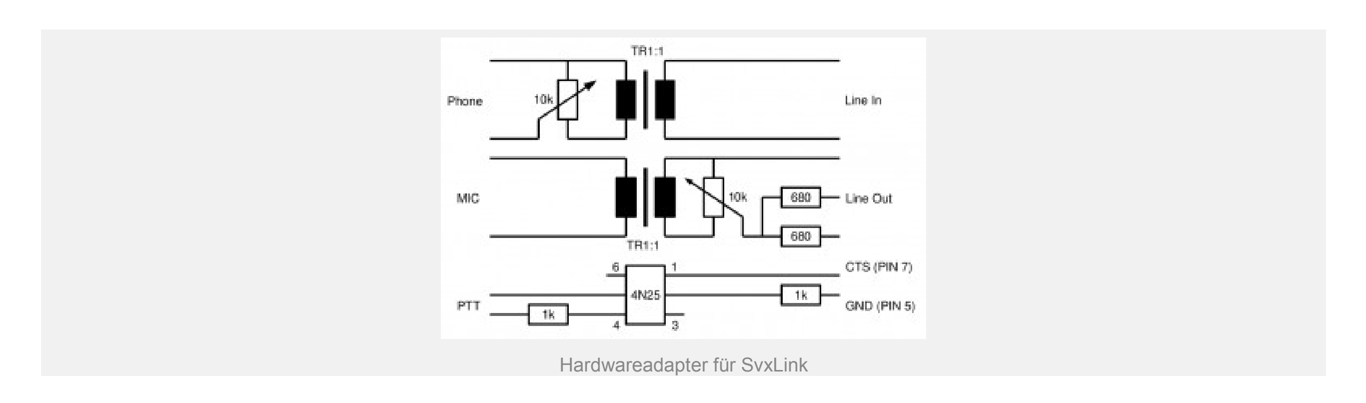
MultiTx

Ein MultiTx-Bereich wird durch die Variablendefinition *TYPE=Multi* eingerichtet. In diesem Bereich muss eine Zuweisung von mindestens einem Sender erfolgen:*TRANSMITTERS=Tx1,Tx2,…*. Die hier definierten Sender werden im Sendefall praktisch gleichzeitig mit dem Audiostream versorgt bzw. angesteuert.

Voter
Ein RX-Voter wird durch die Variablendefinition *TYPE=Voter* eingerichtet. Innerhalb dieses Bereiches muss die Zuweisung mindestens eines Rx als Parameter für RECEIVERS erfolgen:
RECEIVERS=Rx1,NetRx1,…. Zwischen den hier konfigurierten Empfängern erfolgt ein Voting, der Rx mit dem jeweils besten Signal wird "durchgeschaltet", wobei eine Verzögerung um die im *VOTING_DELAY* konfigurierte Zeit [ms] erfolgt.
LocationInfo
Auf diesen Abschnitt wird unter http://svxlink.de/?page_id=325 näher eingegangen.

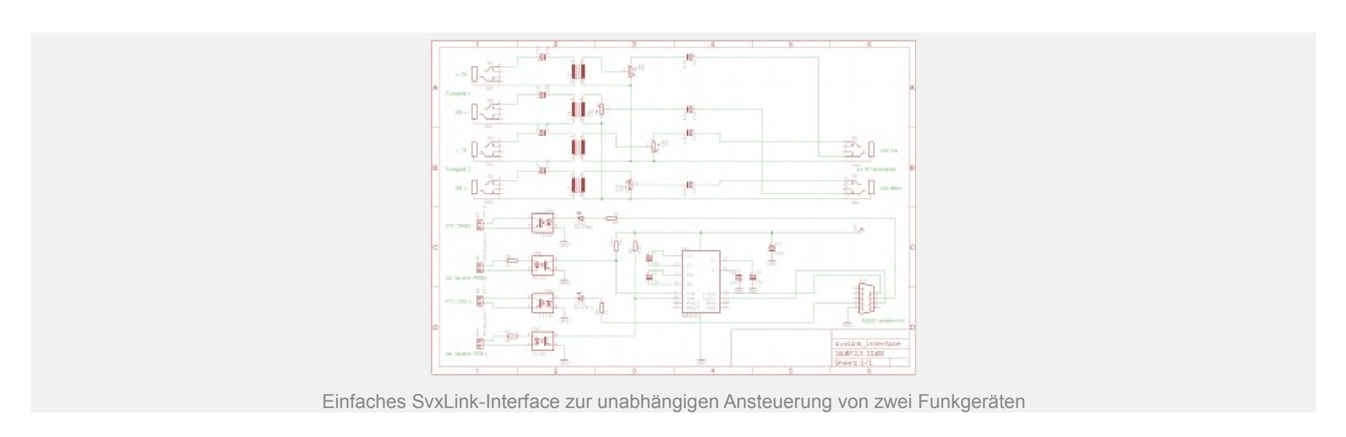
Interfaces

Für SvxLink können die selben Interfaces genutzt werden wie für das original Echolink-Programm für Windows. Wenn man aber selbst basteln möchte, so kann man auch die beiden Kanäle trennen und einen Adapter bauen, mit dem 2 Funkgeräte an eine Soundkarte angeschlossen werden können.
Nachfolgend ein kleiner Schaltplan von Martin (DF1AMB):



Hardwareadapter für SvxLink

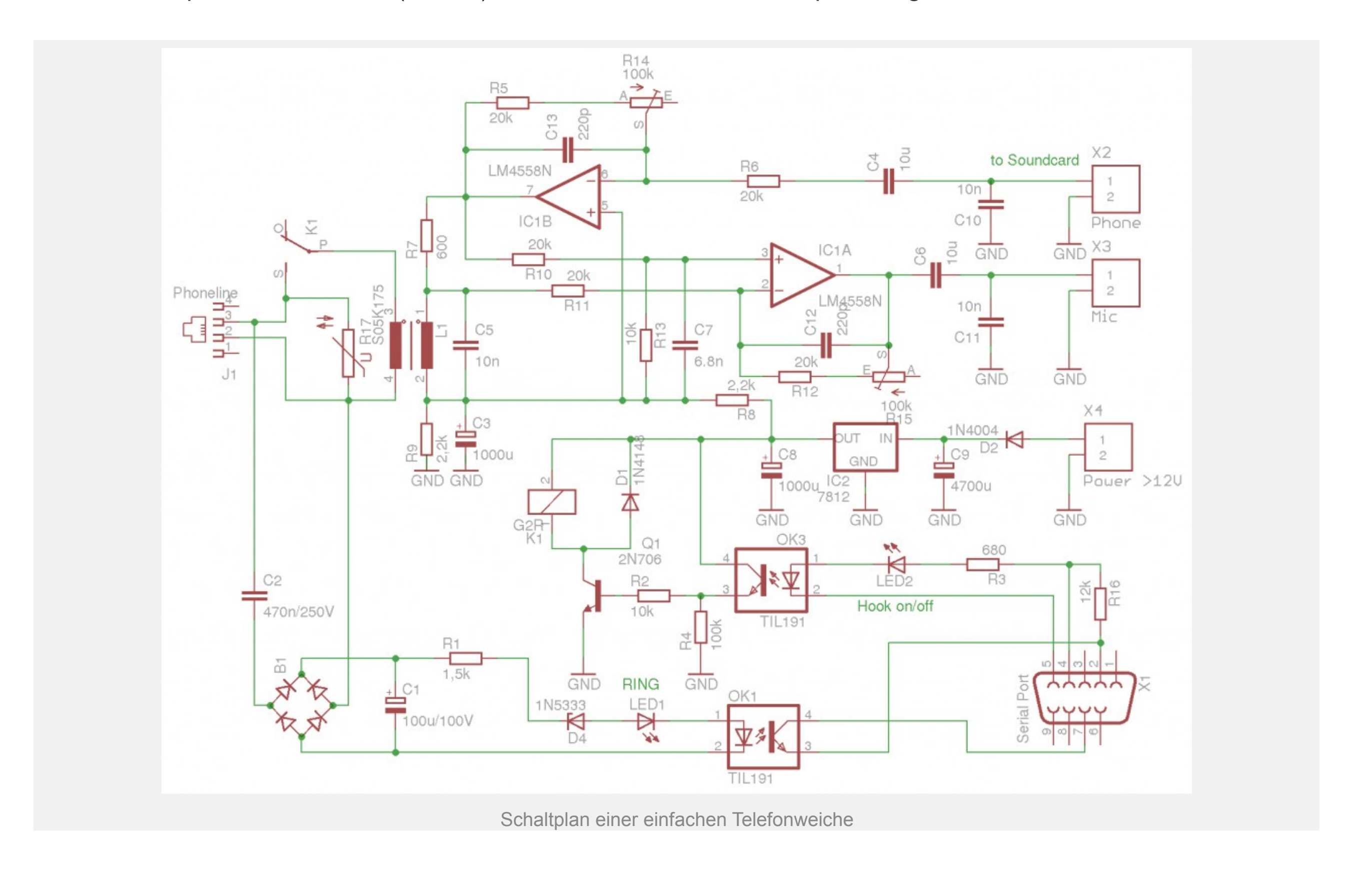
SvxLink kann mit einer Stereo-Soundkarte und einem RS232-Port zwei Funkgeräte unabhängig voneinander ansteuern. Nachfolgend ein einfaches Interface welches auch die Möglichkeit bietet, ein externes Rauschsperrenkriterium (wenn Ihr RX eine entsprechende Information liefert) an SvxLink zu übermitteln.


Einfaches SvxLink-Interface zur unabhängigen Ansteuerung von zwei Funkgeräten

Muss in jedem Fall ein Interface mit galvanischer Trennung verwendetwerden?
Nein, wenn Sie z.B. das Funkgerät an die selbe Spannungsversorgung anschließen mit der der PC betrieben wird, hat die galvanische Trennung ohnehin keine Wirkung. Wir haben über lange Zeit einen Link ohne Trennung betrieben, bei dem das Funkgerät an die 12V des PC's angeschlossen war und hatten keinerlei Probleme.
Ist für SvxLink ein Hardware-DTMF-Dekoder notwendig?
Nein, SvxLink besitzt folgende Softwaredekoder: DTMF, Rufton, CTCSS, SEL5 (ZVEI, EEA,EIA,CCIR,…).

Cradle

Um den PC mit einer analogen Telefonleitung zu verbinden gibt es verschiedene Möglichkeiten. Eine davon ist eine sehr simple Telefonweiche (Cradle). Im ersten Bild ist der Schaltplan dargestellt.



Schaltplan einer einfachen Telefonweiche

Und hier das Layout für das Board.

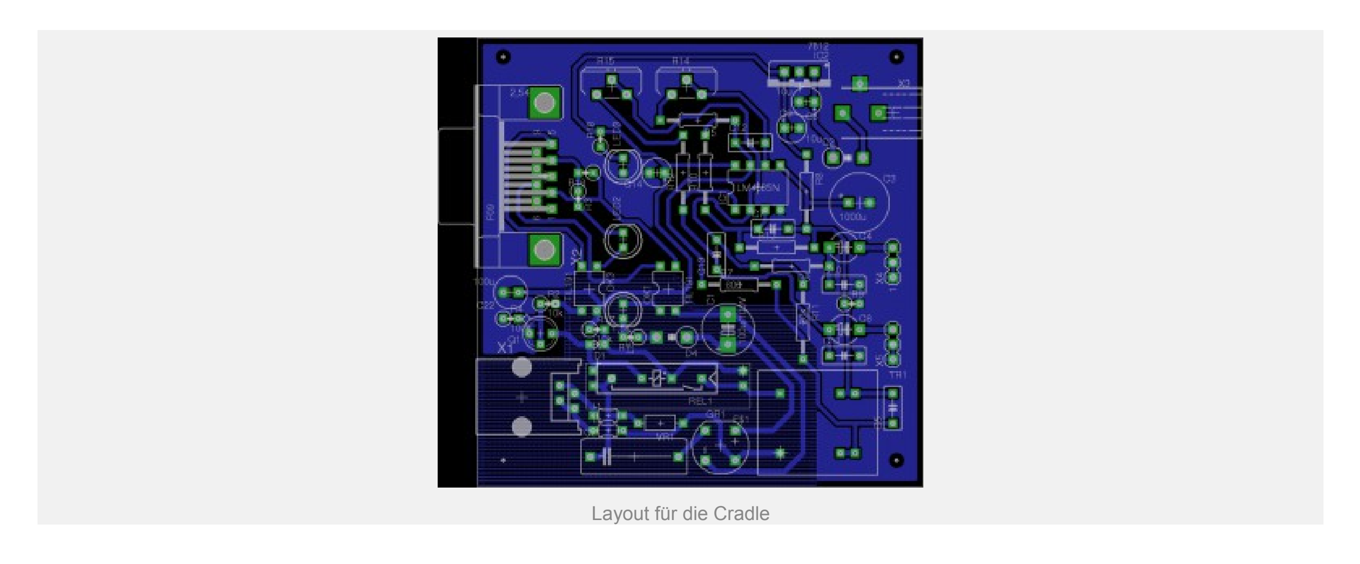
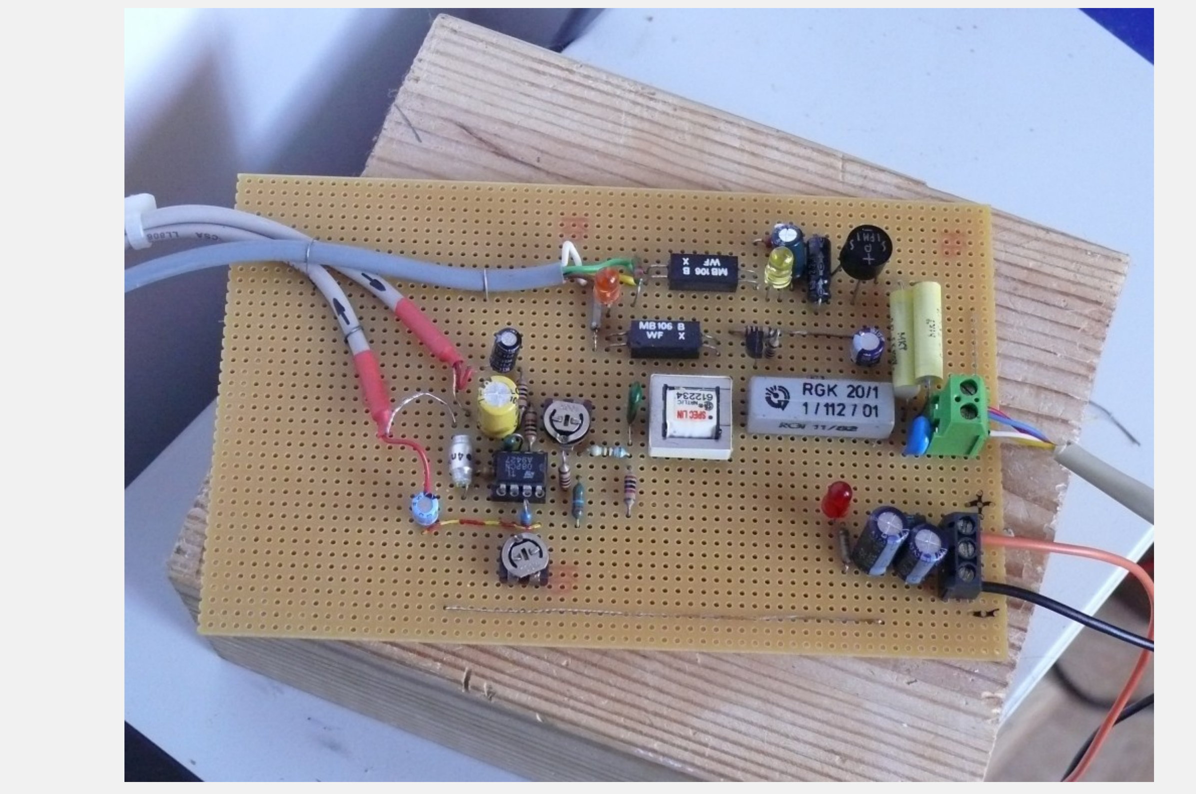

Layout für die Cradle

Bild des Testaufbaus.



Testaufbau der Cradle ... Tja, ich bin eben ein Ossi ;-)

Analoges Voicemodem

Die Ansteuerung des analogen Voicemodems wird über eine normale RS232 vorgenommen. Audioein- und Ausgang können direkt mit Mic- bzw. Lineout der Soundkarte verbunden werden. Es funktioniert ausdrücklich nur mit einem analogen Voicemodem, dieses muss die AT-Befehle: AT+VLS=…, AT+CLS=… bzw. AT#VLS=…, AT#CLS= unterstützen. Leider kocht auch hier jeder Hersteller sein eigenes Süppchen und das Verhalten der Modems bei ein und den selben Befehlen ist oft sehr unterschiedlich.



Creatix VF14400

Getestet haben wir mit einem uralten Creatix VF14400. Das eingebaute Elektret-Mikrofon wurde durch eine Buchse ersetzt, damit man es mit dem Lautsprecherausgang der Soundkarte verbinden kann.
Weitere Information bzgl. Installation und Konfiguration folgen in Kürze.

GSM-Interface

Für Mobiltelefone der Siemens M/S-35-Serie ist eine kleine Zusatzschaltung erforderlich. Ein Schaltplan ist hier zu finden. Das größte Problem ist eigentlich einen passenden Stecker zu bekommen, bei dem alle notwendigen Kontakte verfügbar sind. Auf ebay gibt es leider nur Adapter, die entweder die Audioleitungen herausführen ODER eine Datenkommunikation mit/ohne Akkumulatorladefunktion bereitstellen….leider nicht alles.



GSM-Interface für Mobiltelefone der Siemens-Serie (C35, M35)

Steckerbelegungen von Mobiltelefonen finden Sie auf Nobbi.com
Etwas einfacher gestaltet sich der Anschluß eines Sony-Ericsson P910i, hier ist kein Interface mit MAX3232 erforderlich.



GSM-Telefon (hier Sony Ericsson P910i) zum Anschluß an SvxLink

Die billigen Soundkarten für teilweise 5€ eignen sich recht gut dafür. Allerdings besitzen diese oft keinen Stereo-Mikrofoneingang.
Momentan befindet sich alles im absoluten Bastelstadium. Getestet wurde mit einem Sony Ericsson P910i, es müssten allerdings auch andere Geräte wie z.B. K610, 750 funktionieren. Eine Unterstützung für Siemens C35 etc. ist in Arbeit.
Momentan muss das Gerät auf den AT-Befehl "AT&FE0*ECAM=1″ mit OK reagieren. Sonst funktioniert es nicht.
Für die Unterstützung weiterer Geräte benötigen wir eine AT-Befehlsreferenz zum interessierenden Gerät. Wer etwas entsprechendes besitzt und es zur Verfügung stellen möchte, bitte melden.
Gesucht werden momentan AT-Befehlssätze zu:

Blackberry 9000

bold 9000

Locale-Konzept

Da es immer wieder zu Fragen und Problemen führt, wird auf dieser Seite das SvxLink-locale-Konzept etwas genauer beschrieben. Der Hintergrund des locale-Konzeptes ist einerseits den Usern eine einfache Anpassung an die in der jeweiligen Landessprache üblichen Ausdrucksweise zu ermöglichen. Andererseits sollen diese userspezifischen Änderungen nicht bei Systemupdates jedes Mal neu überschrieben werden sondern natürlich auch erhalten bleiben. Es ist deshalb ungünstig die mitinstallierten tcl-Dateien nach den eigenen Bedürfnissen zu verändern, da diese Änderungen nach jedem Update wieder verloren gehen würden. Auch ein Sichern der tcl-Dateien und späteres Rücksichern ist ungünstig, da die neuen tcl-Dateien eventuell neue oder veränderte Prozeduren enthalten könnten, die dann durch die alten Einstellungen überschrieben würden.

Beim Start von SvxLink wird normalerweise vom Eventhandler zunächst die Datei*events.tcl* geladen, welche die anderen modul- und logikspezifischen tcl-Dateien nachlädt. Das passiert in der folgenden Reihenfolge (Stand 1/2012 für deutsche Umgebung):
1) /usr/share/svxlink/events.d/
2) /usr/share/svxlink/events.d/local/
3) /usr/share/svxlink/sounds/de_DE/events.d/
4) /usr/share/svxlink/sounds/de_DE/events.d/local/
5) /usr/share/svxlink/modules.d/

Wenn man also eine tcl-Systemfunktion mit einer eigenen Funktion überschreiben möchte um z.B. die Ansage der Uhrzeit an die deutsche Sprechweise anzupassen, so erstellt man jeweils in den zugehörigen local-Verzeichnissen eine beliebige tcl-Datei, definiert darin zunächst den Namespace und innerhalb dieses die eigene Funktion. Da diese eigene Datei im *local*-Verzeichnis NACH den System-tcl's gelesen wird, werden die System-tcl-Funktionen durch die selbstdefinierten Funktionen überschrieben.
Hier mal Beispiel für die Änderung des roger beeps, dieser wird in der Datei Logic.tcl generiert. Im Beispiel soll die Tonhöhe von 440Hz auf 800Hz verändert werden:

1) Erstellen Sie eine Datei *beep.tcl* mit folgendem Inhalt:

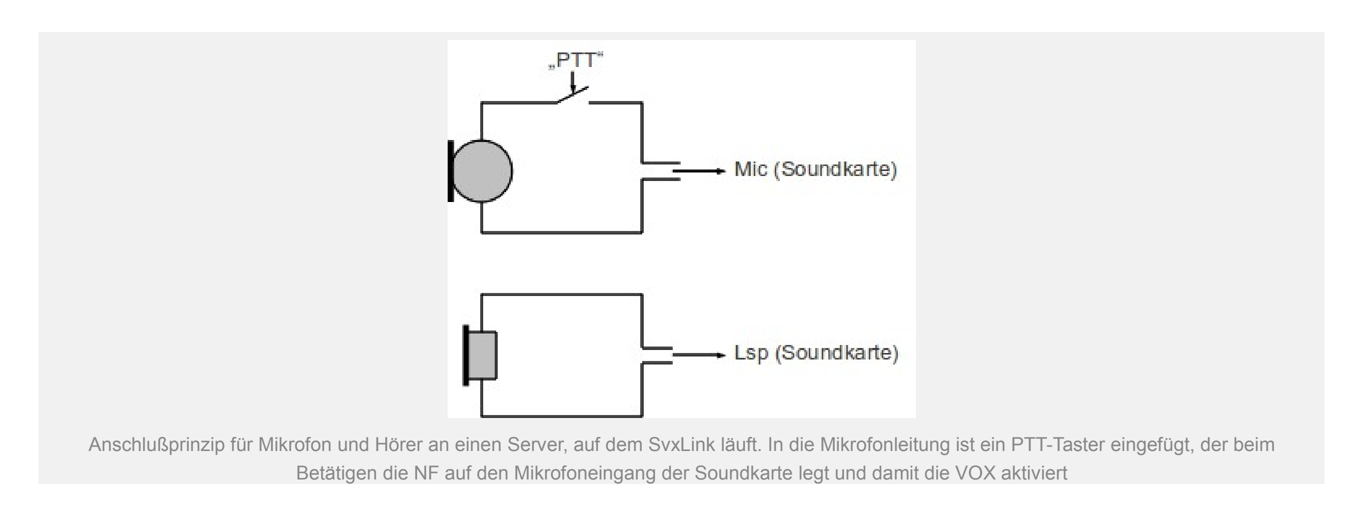
```
namespace eval Logic {
#
# Executed when the squelch just have closed and the RGR_SOUND_DELAY timer has
# expired.
#
proc send_rgr_sound {} {
  playTone 800 80 50;
  playSilence 50;
  }
}
```

2) Speichern Sie diese Datei in dem Verzeichnis*/usr/share/svxlink/sounds/de_DE/events.d/local/* ab. Starten Sie SvxLink neu, damit die neuen Einstellungen geladen werden können.

Tipps und Tricks

MicSpkr-Logic

Die “MicSpkr-Logic” ist eigentlich nur eine “verbogene” Repeater- oder SimplexLogic, die den Anschluß einer “Hör-Sprech-Garnitur” an eine Relaisstelle (auch Simplex-Link) ermöglicht. Damit ergibt sich die Möglichkeit direkt an der Kommunikation per Funk teilzunehmen. Sie wird dabei gleichberechtigt zu den übrigen Logiken konfiguriert und kann entsprechend genutzt werden.



Anschlußprinzip für Mikrofon und Hörer an einen Server, auf dem SvxLink läuft. In die Mikrofonleitung ist ein PTT-Taster eingefügt, der beim Betätigen die NF auf den Mikrofoneingang der Soundkarte legt und damit die VOX aktiviert

Merkmale
* es wird ein Soundkartenkanal benötigt (Lautsprecherausgang und Mikrofoneingang)
* Konfiguration als Repeater- oder SimplexLogic
* Squelch sollte VOX sein, also sprachgesteuert, so spart man eine serielle Schnittstelle für die PTT
* TYPE=Repeater ermöglicht auch einen quasi-Vollduplexbetrieb

Beispielkonfiguration, Ausschnitt aus der /etc/svxlink/svxlink.conf

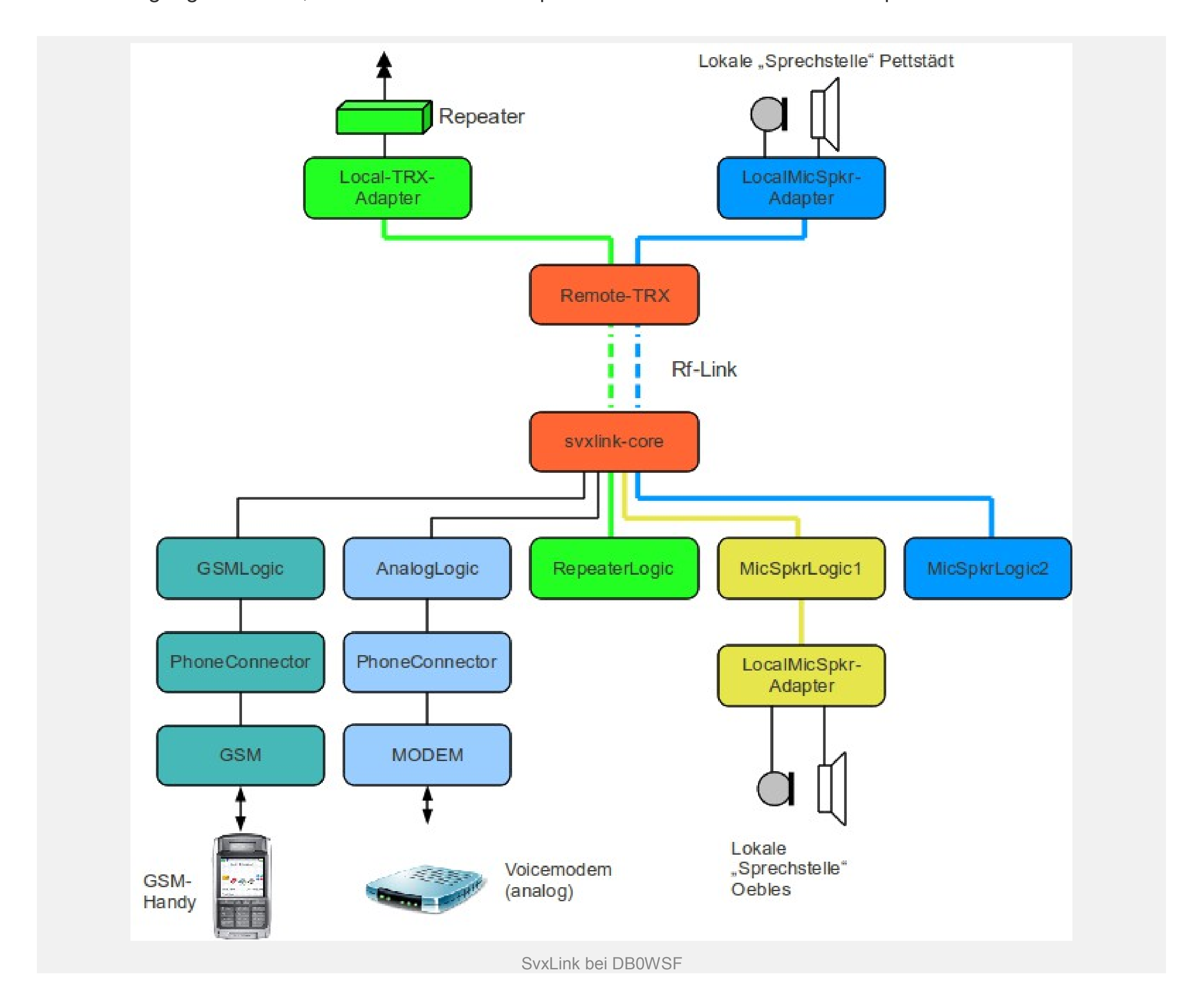
```
[GLAOBAL]
LOGICS=...,MicSpkrLogic,...
[MicSpkrLogic]
TYPE=Simplex
RX=LocalMic
TX=LocalSpkr
DEFAULT_LANG=de_DE
CALLSIGN=DL1HRC
EVENT_HANDLER=/usr/share/svxlink/events.tcl
[LocalMic]
TYPE=Local
AUDIO_DEV=oss:/dev/dsp
AUDIO_CHANNEL=0
SQL_DET=VOX
SQL_START_DELAY=0
SQL_DELAY=0
SQL_HANGTIME=500
SQL_TIMEOUT=600
VOX_FILTER_DEPTH=20
VOX_THRESH=1000
[LocalSpkr]
TYPE=Local
AUDIO_DEV=oss:/dev/dsp
AUDIO_CHANNEL=0
```

PTT_PORT=NONE

PTT_PIN=!DTRRTS

Für Erklärungen zu den Konfigurationsvariablen siehe Repeater- oder Simplexlogic.

Als kleines Beispiel noch eine Übersicht, wie es bei DB0WSF eingerichtet und betrieben wird. Es soll nur eine kleine Anregung darstellen, selber mal damit zu experimentieren. Es macht wirklich Spaß.



SvxLink bei DB0WSF

Mehrere Qtel-Clients mit SvxLink (lokal)

Mehrere Qtel-Clients an einem DSL-Anschluß betreiben

Mit SvxLink ist es relativ einfach möglich, mehrere PCs mit einem Echolink-Client (Qtel) an einem DSL-Anschluß zu betreiben um z.B. von verschiedenen Räumen (Shack, Küche,Wohnzimmer,…) aus gleichzeitig QRV zu sein. Ein kleiner Wermutstropfen soll gleich am Anfang nicht verschwiegen werden. Es geht nur eine gleichzeitige Echolink-Verbindung mit Stationen von/nach außen. Das Führen von mehreren unabhängigen Gesprächen ist nicht möglich was allerdings prinzipbedingt ist.

Die folgende Grafik zeigt eine mögliche Konfiguration.

Echolink-
Verzeichnisdienst
Internet
z.B. dl0abc.dyndns.org
DSL-Router
DLOABC-L
PC1: SvxLink-Server
IP: 192.168.0.1
qtel1
Call: DLOABC
IP: 192.168.0.10
qtel 2
Call: DL5XXX
IP: 192.168.0.11
qtel 3
Call: DL2YYY
IP: 192.168.0.12

SvxLink: Mehrere QTel-Clients an einem DSL-Anschluß

Auf PC1 ist ein SvxLink-Server installiert und am Echolink-Netzwerk unter z.B. DL1ABC-L angemeldet. Auf den anderen PCs läuft nur der Client Qtel. Die einzelnen Qtel-Clients können nun so konfiguriert werden, als würden sie sich eigenständig im Echolink-Netzwerk anmelden was sie auch tun. Jede der Stationen (DL0ABX-L, DL0ABC,DL5XXX und DL2YYY) ist nun für alle Echolink-Teilnehmer weltweit zu sehen.

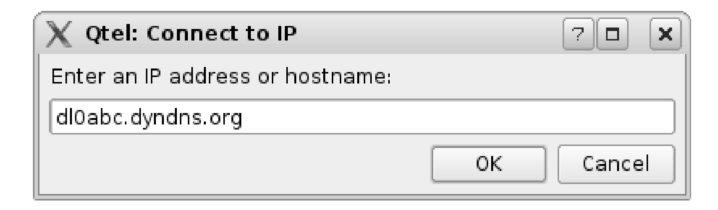
Die Clienten verbinden sich nun über das private Netzwerk mit dem eigenen SvxLink-Server DL0ABC-L.



Wird nun einer der genannten Stationen von einem Echolink-Teilnehmer außerhalb des eigenen Netzwerkes connected, so nimmt grundsätzlich DL0ABC-L die Verbindung entgegen. Da aber auch die anderen drei Stationen (DL0ABC, DL5XXX sowie DL2YYY) mit DL0ABC-L connected sind, werden alle eingehenden

Audiodaten an alle weitergeleitet. Wenn ein Teilnehmer des eigenen Home-Netzwerkes spricht, z.B. DL2YYY, so hören alle anderen mit (DL0ABC-L, DL0ABC, DL5XXX und die aus dem Internet verbundene Station). Im oben gezeigten Fall wird das Gespräch also auch per Funk ausgesendet und Funkteilnehmer können sich in das laufende Gespräch hineinmelden.
Auch ohne Connect von außen ist eine Verbindung zwischen den Clients möglich, aber Achtung wenn man ein eigenes Relais oder einen eigenen Link betreibt. Das Gespräch wird automatisch ausgesendet!

Auch Qtel-Stationen außerhalb des eigenen Netzwerkes können sich direkt mit dem eigenen SvxLink-Server verbinden ohne dabei die Verbindung über den Echolink-Verzeichnisdienst herstellen zu müssen:



Hierbei muß der eigene SvxLink-Server allerdings bei einem DDNS-Anbieter wie z.B. dyndns.org registriert sein, da es sonst sehr schwierig wird bei ständig wechselnden IP-Adressen von außen den SvxLink-Server zu erreichen. Die externe Station muß aber zwingend im Echolink-Netzwerk angemeldet und authentifiziert sein! SvxLink prüft beim Connect ab, ob es sich um eine im Echolink-Netzwerk registrierte und authentifizierte Station handelt.
Wer das einmal ausprobieren möchte, dem sei Puppylink empfohlen, ein Linux-Livesystem, welches von CD gebootet wird und Qtel bereits beinhaltet.

Dieser Anordnung sind nur durch die Bandbreite des DSL-Anschlusses bzw. des eigenen Netzwerkes Grenzen gesetzt.
Damit SvxLink die Anfragen der eigenen Clients akzeptiert, muß dies in der Datei /etc/svxlink.d/ModuleEchoLink.conf explizit freigeschaltet werden:

```
[ModuleEchoLink]
...
ALLOW_IP=192.168.0.0/24
...
```

wobei natürlich der IP-Adressraum des eigenen privaten Netzwerksegmentes eingetragen werden muß.

Linking RemoteTrx und SvxLink

Bei dieser Konfiguration werden von einem Remotetrx/SvxLink-Basis empfangenen Signale an die angeschlossenen Stationen übertragen und dort ausgegeben. Ebenfalls können die angeschlossenen Stationen am EchoLink-Betrieb teilnehmen bzw. entsprechende EchoLink-Verbindungen auf- und abbauen. Die hier dargestellte Variante nutzt sendeseitig einen *MultiTx* und empfangsseitig die in SvxLink eingebaute *Voter*-Funktion.

Auf der **SvxLink-Basisstation läuft die Applikation *svxlink***, auf den**RemoteTrx(en) die Applikation *remotetrx*** (bei SvxLink mit dabei). Der Verbindungsaufbau erfolgt dabei grundsätzlich von der Svxink-Basisstation zu den RemoteTrx(en). Achten Sie darauf, dass diese über die IP-Adresse (Achtung bei dhcp) oder per Hostnamen (außerhalb des eigenen Netzes -> dyndns, o.ä) auf dem konfigurierten Port erreichbar sind.

Als Codec kann innerhalb des eigenen Netzes entweder ein RAW-Stream oder der S16-Codec verwendet werden, bei Verbindungen ins Internet über eine DSL-Verbindung sollte man auf den SPEEX und bei geringeren Qualitätsansprüchen auf den GSM-Codec zurückgreifen.

Die Nachteile dieser Konfigurationsvariante sollen aber nicht verschwiegen werden:

- Die Verbindungen zwischen den Stationen sind statisch, also nicht steuerbar (Trennen,Verbinden auf Kommando)
- keine spezifischen Logic-Funktionen für die einzelnen RemoteTrx(e) verfügbar (Metar,Parrot,…)
- Voterfunktionen kosten Zeit, d.h. höhere Latenzen

Eine Weiterentwicklung stellt das LogicLinking dar, welches es das gezielte Verbinden/Trennen einzelner Teilnehmer sowie Timeout-Funktionen ermöglichen.

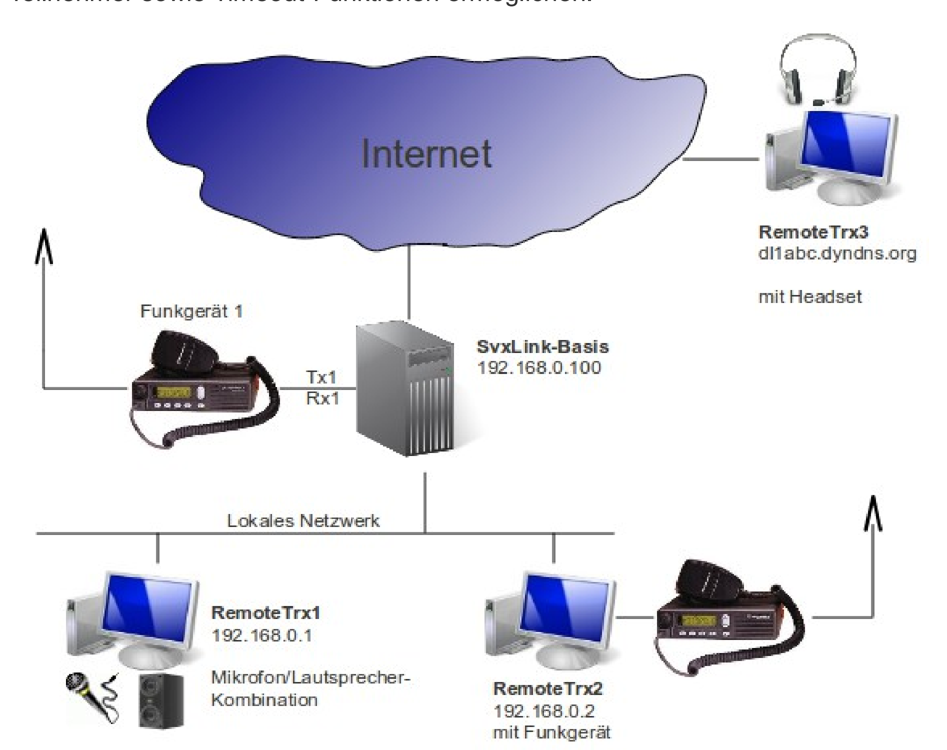


Abbildung 1: Prinzip SvxLink und RemoteTrx(e)

Die nachfolgende Abbildung stellt die einzelnen Konfigurationsabschnitte der SvxLink-Basis und eines RemoteTrx’s deutlich(er) heraus.

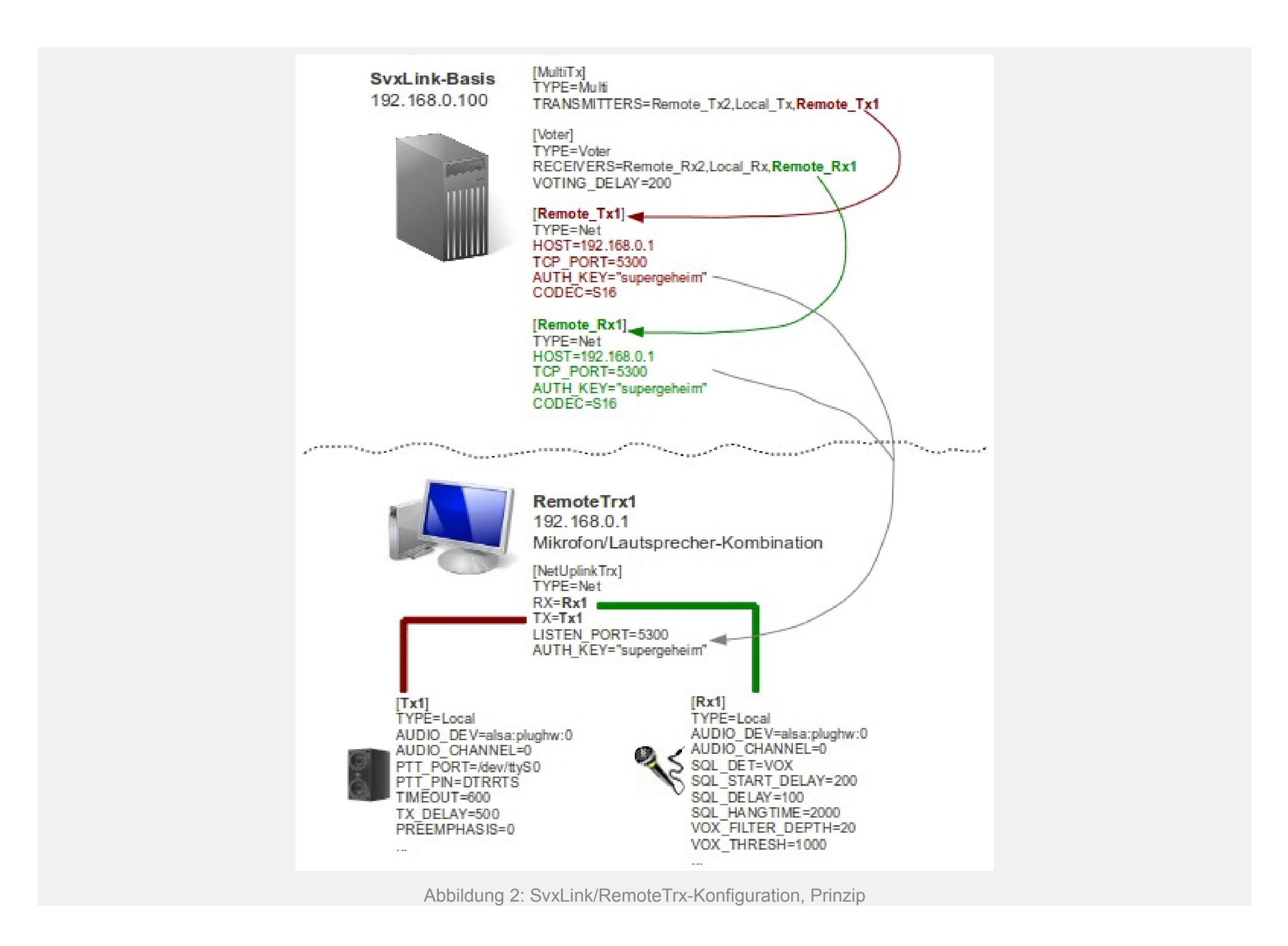


Abbildung 2: SvxLink/RemoteTrx-Konfiguration, Prinzip

Nachfolgend die Konfigurationsdateien für die einzelnen Stationen analog zur Abbildung 1.

SvxLink (Basisstation), Konfigurationsdatei svxlink.conf

```
# Konfigurationsdatei svxlink.conf auf der lokalen SvxLink-Basis

[GLOBAL]

MODULE_PATH=/usr/lib/svxlink

LOGICS=RepeaterLogic

CFG_DIR=svxlink.d

TIMESTAMP_FORMAT="%d.%m.%Y %H:%M:%S"

CARD_SAMPLE_RATE=48000



[RepeaterLogic]

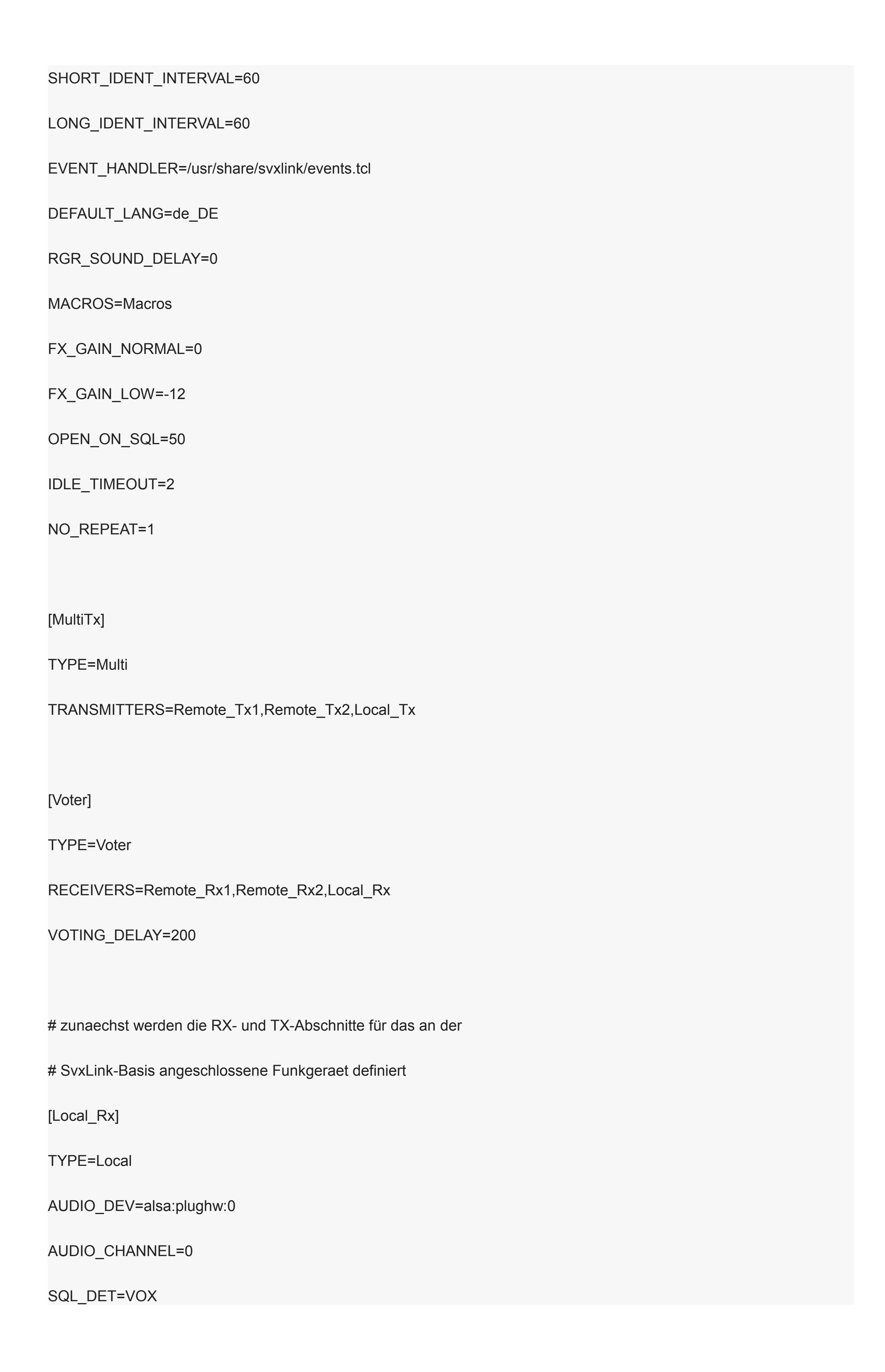
TYPE=Repeater

RX=Voter

TX=MultiTx

MODULES=ModuleHelp,ModuleParrot,ModuleEchoLink,ModuleTclVoiceMail

CALLSIGN=DL1ABC
```



```
SHORT_IDENT_INTERVAL=60

LONG_IDENT_INTERVAL=60

EVENT_HANDLER=/usr/share/svxlink/events.tcl

DEFAULT_LANG=de_DE

RGR_SOUND_DELAY=0

MACROS=Macros

FX_GAIN_NORMAL=0

FX_GAIN_LOW=-12

OPEN_ON_SQL=50

IDLE_TIMEOUT=2

NO_REPEAT=1


[MultiTx]

TYPE=Multi

TRANSMITTERS=Remote_Tx1,Remote_Tx2,Local_Tx


[Voter]

TYPE=Voter

RECEIVERS=Remote_Rx1,Remote_Rx2,Local_Rx

VOTING_DELAY=200


# zunaechst werden die RX- und TX-Abschnitte für das an der

# SvxLink-Basis angeschlossene Funkgeraet definiert

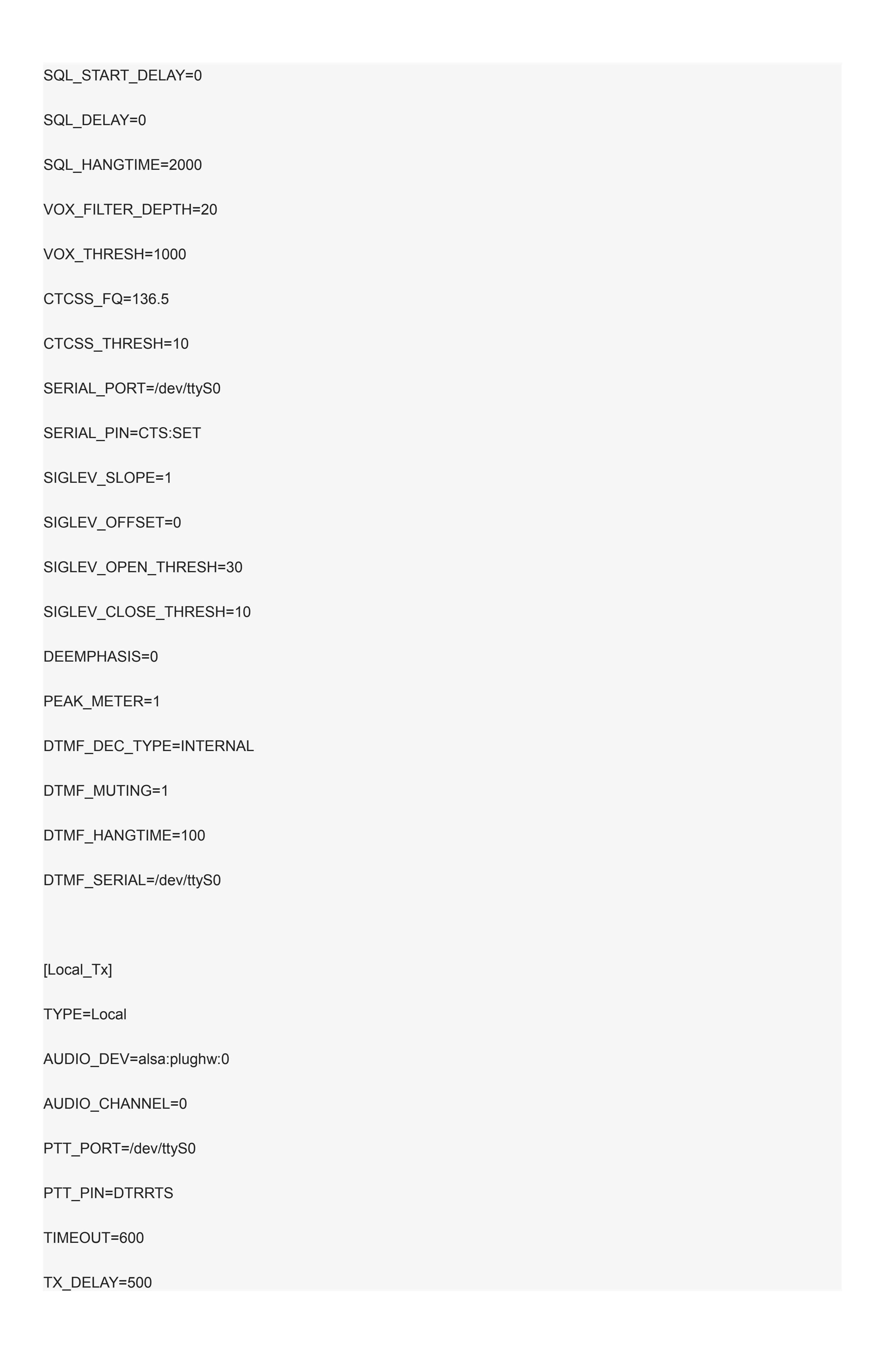
[Local_Rx]

TYPE=Local

AUDIO_DEV=alsa:plughw:0

AUDIO_CHANNEL=0

SQL_DET=VOX
```



```
SQL_START_DELAY=0

SQL_DELAY=0

SQL_HANGTIME=2000

VOX_FILTER_DEPTH=20

VOX_THRESH=1000

CTCSS_FQ=136.5

CTCSS_THRESH=10

SERIAL_PORT=/dev/ttyS0

SERIAL_PIN=CTS:SET

SIGLEV_SLOPE=1

SIGLEV_OFFSET=0

SIGLEV_OPEN_THRESH=30

SIGLEV_CLOSE_THRESH=10

DEEMPHASIS=0

PEAK_METER=1

DTMF_DEC_TYPE=INTERNAL

DTMF_MUTING=1

DTMF_HANGTIME=100

DTMF_SERIAL=/dev/ttyS0


[Local_Tx]

TYPE=Local

AUDIO_DEV=alsa:plughw:0

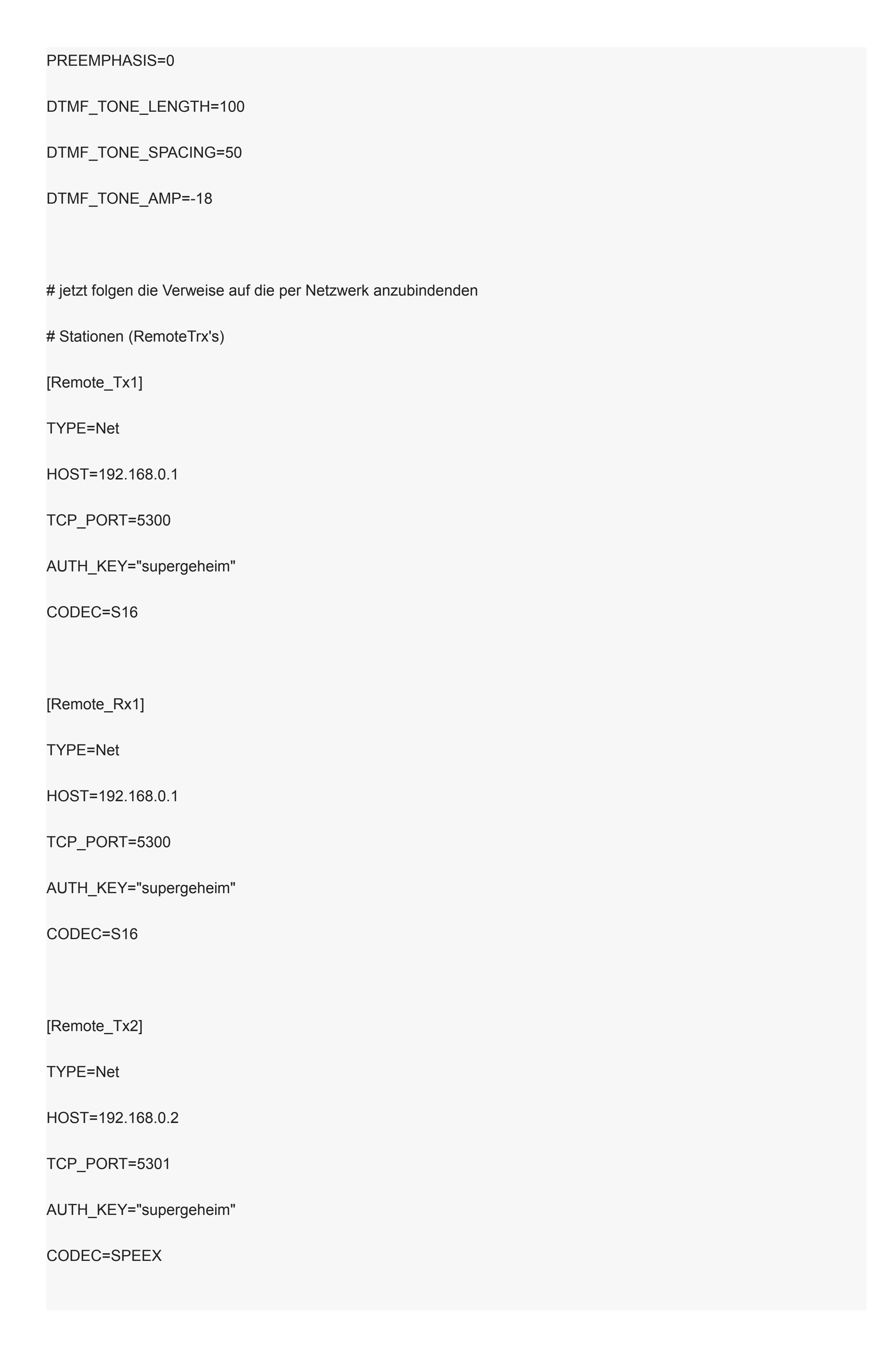
AUDIO_CHANNEL=0

PTT_PORT=/dev/ttyS0

PTT_PIN=DTRRTS

TIMEOUT=600

TX_DELAY=500
```



```
PREEMPHASIS=0

DTMF_TONE_LENGTH=100

DTMF_TONE_SPACING=50

DTMF_TONE_AMP=-18


# jetzt folgen die Verweise auf die per Netzwerk anzubindenden

# Stationen (RemoteTrx's)

[Remote_Tx1]

TYPE=Net

HOST=192.168.0.1

TCP_PORT=5300

AUTH_KEY="supergeheim"

CODEC=S16


[Remote_Rx1]

TYPE=Net

HOST=192.168.0.1

TCP_PORT=5300

AUTH_KEY="supergeheim"

CODEC=S16


[Remote_Tx2]

TYPE=Net

HOST=192.168.0.2

TCP_PORT=5301

AUTH_KEY="supergeheim"

CODEC=SPEEX
```

```
[Remote_Rx2]

TYPE=Net

HOST=192.168.0.2

TCP_PORT=5301

AUTH_KEY="supergeheim"

CODEC=SPEEX
```

Nachfolgend die Konfigurationen für die RemoteTrx'e.

RemoteTrx1 (auf 192.168.0.1), Konfigurationsdatei remotetrx.conf:

```
# Konfigurationsdatei remotetrx.conf auf dem ersten RemoteTrx

[GLOBAL]

TRXS=NetUplinkTrx

TIMESTAMP_FORMAT="%c"

CARD_SAMPLE_RATE=48000



[NetUplinkTrx]

TYPE=Net

RX=Rx1

TX=Tx1

LISTEN_PORT=5300

FALLBACK_REPEATER=0

AUTH_KEY="supergeheim"



[Rx1]

TYPE=Local

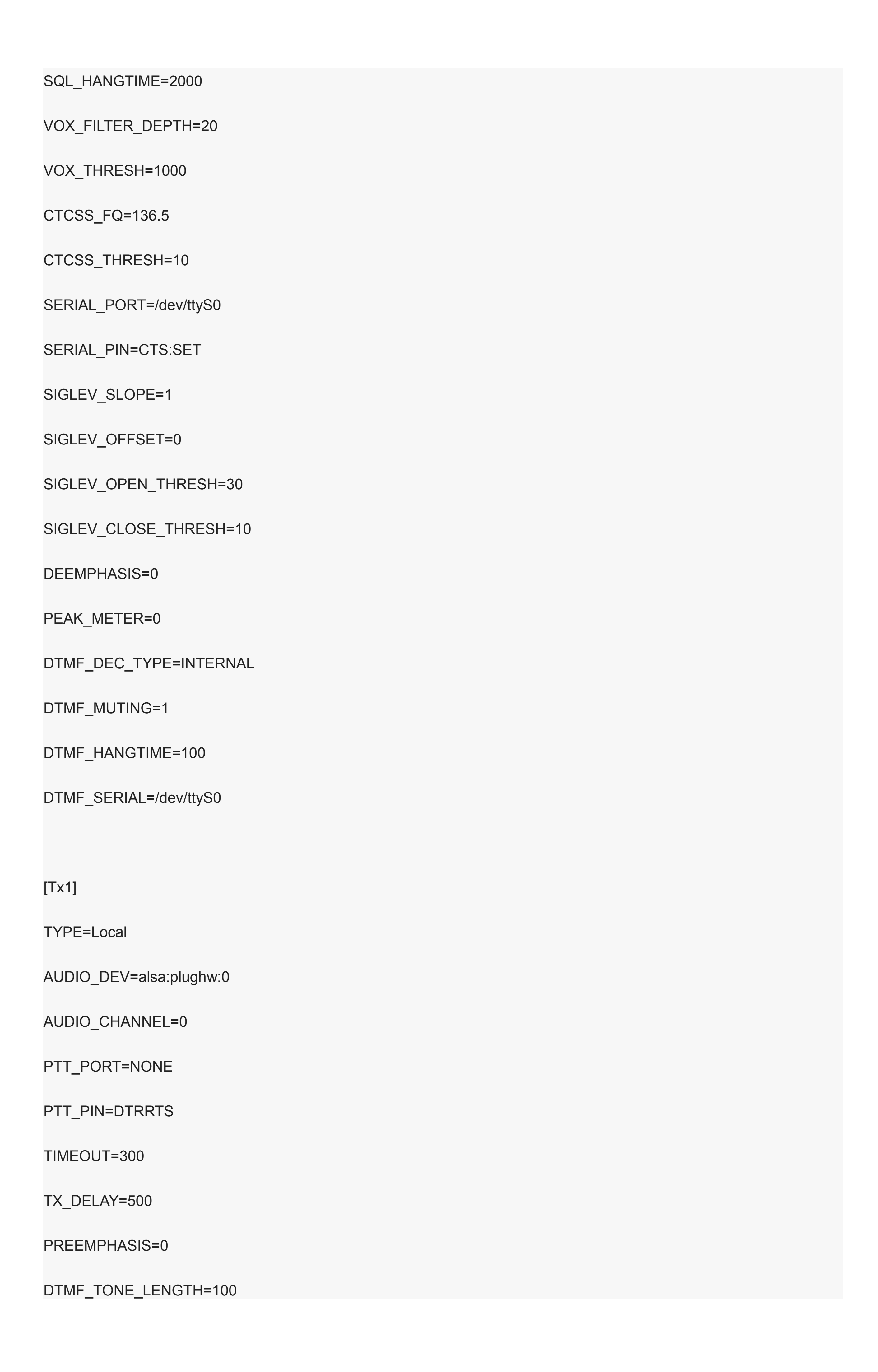
AUDIO_DEV=alsa:plughw:0

AUDIO_CHANNEL=0

SQL_DET=VOX

SQL_START_DELAY=200

SQL_DELAY=100
```



```
SQL_HANGTIME=2000

VOX_FILTER_DEPTH=20

VOX_THRESH=1000

CTCSS_FQ=136.5

CTCSS_THRESH=10

SERIAL_PORT=/dev/ttyS0

SERIAL_PIN=CTS:SET

SIGLEV_SLOPE=1

SIGLEV_OFFSET=0

SIGLEV_OPEN_THRESH=30

SIGLEV_CLOSE_THRESH=10

DEEMPHASIS=0

PEAK_METER=0

DTMF_DEC_TYPE=INTERNAL

DTMF_MUTING=1

DTMF_HANGTIME=100

DTMF_SERIAL=/dev/ttyS0


[Tx1]

TYPE=Local

AUDIO_DEV=alsa:plughw:0

AUDIO_CHANNEL=0

PTT_PORT=NONE

PTT_PIN=DTRRTS

TIMEOUT=300

TX_DELAY=500

PREEMPHASIS=0

DTMF_TONE_LENGTH=100
```

```
DTMF_TONE_SPACING=50

DTMF_TONE_AMP=-18
```

RemoteTrx2 (auf 192.168.0.2), Konfigurationsdatei remotetrx.conf:

```
# Konfigurationsdatei remotetrx.conf auf dem zweiten RemoteTrx

[GLOBAL]

TRXS=NetUplinkTrx

TIMESTAMP_FORMAT="%c"

CARD_SAMPLE_RATE=48000



[NetUplinkTrx]

TYPE=Net

RX=Rx1

TX=Tx1

LISTEN_PORT=5301

FALLBACK_REPEATER=0

AUTH_KEY="supergeheim"



[Rx1]

TYPE=Local

AUDIO_DEV=alsa:plughw:0

AUDIO_CHANNEL=0

SQL_DET=VOX

SQL_START_DELAY=200

SQL_DELAY=100

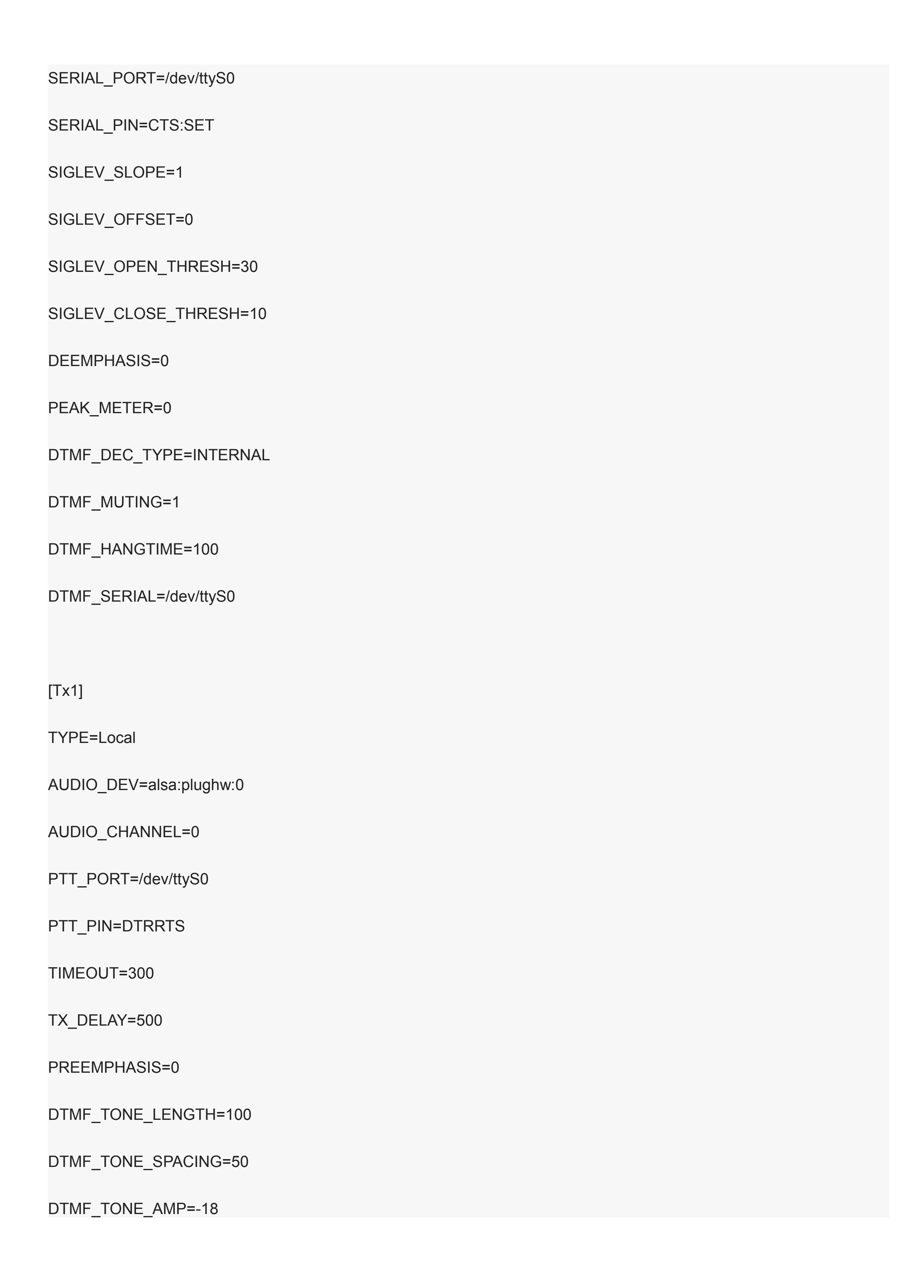
SQL_HANGTIME=2000

VOX_FILTER_DEPTH=20

VOX_THRESH=1000

CTCSS_FQ=136.5

CTCSS_THRESH=10
```



```
SERIAL_PORT=/dev/ttyS0

SERIAL_PIN=CTS:SET

SIGLEV_SLOPE=1

SIGLEV_OFFSET=0

SIGLEV_OPEN_THRESH=30

SIGLEV_CLOSE_THRESH=10

DEEMPHASIS=0

PEAK_METER=0

DTMF_DEC_TYPE=INTERNAL

DTMF_MUTING=1

DTMF_HANGTIME=100

DTMF_SERIAL=/dev/ttyS0


[Tx1]

TYPE=Local

AUDIO_DEV=alsa:plughw:0

AUDIO_CHANNEL=0

PTT_PORT=/dev/ttyS0

PTT_PIN=DTRRTS

TIMEOUT=300

TX_DELAY=500

PREEMPHASIS=0

DTMF_TONE_LENGTH=100

DTMF_TONE_SPACING=50

DTMF_TONE_AMP=-18
```

Internetstreaming

Hier ist beschrieben, wie man den Funkverkehr des eigenen Repeater/Link im Internet veröffentlichen kann.

Internetstreaming vor Linux Kernel Version 2.6.37

Um den Funkverkehr einer breiten Masse zugänglich zu machen bietet sich die Einrichtung eines Streamingservers an.
Im Folgenden wird die Einrichtung beschrieben. Diese ist momentan aber noch nicht optimal, an einer Verbesserung wird gearbeitet. Das Problem ist, dass Ices2 offensichtlich nicht von einem Playback-Device streamen kann. Aus diesem Grund ist momentan bei dieser Version noch eine zusätzliche Soundkarte erforderlich. Es ist aber mein Ziel, darauf zukünftig verzichten zu können.

Voraussetzungen:
* Icecast2 als Streamingserver
* Ices2 als Streaming client
* eine einfache USB-Soundkarte, bei der der Lautsprecherausgang und Mikrofoneingang miteinander verbunden werden.

Ich nutze hierfür eine billige USB-Soundkarte für 4,95 € bei einem bekannten virtuellen Auktionshaus.
Zunächst müssen Icecast2 und Ices2 installiert und nach Wunsch konfiguriert werden, im Internet gibt es hierfür mehr als genügend Konfigurationsbesispiele.
Der ices2 wird auf dem selben PC installiert, auf dem auch SvxLink läuft. Hier gibt es folgendes zu beachten:

Ices2 kann nicht auf das selbe Device zugreifen, welches von SvxLink als Sounddevice genutzt wird, es kommt hierbei zu einer “resource busy” Meldung.
Aus diesem Grund splittet man den Audiopfad mit Hilde der */etc/asound.conf* auf:

```
pcm.convert {
type plug
slave { pcm "hw:1,0" }
}
pcm.dmix1 {
type copy
slave { pcm convert }
}
pcm.dmix2 {
type copy
slave { pcm dmix1 }
}
```

Ices2 grabbt nun von z.B. dmix2 und SvxLink liefert die Daten an dmix1. Nachfolgend Ausschnitte für beide Applikationen:

/etc/ices2.xml:

```
...
< stream >
< metadata >
< name >DB0WSF< /name>
< genre >Hamradio Network< /genre >
< description >Livestream of DB0WSF-R / Pettstaet / JO51wg< /description >
< url >http://svxlink.de< /url >
< /metadata >
< input >
```

```
< module >alsa< /module >
< param name="rate">16000< /param>
< param name="channels">1< /param>
< param name="device">plug:dmix2< /param>
< param name="buffer-time">500< /param>
< param name="metadata">1< /param>
< param name="metadatafilename">/tmp/info.txt< /param>
< /input>
...
```

SvxLink wird als Multi-Tx konfiguriert (*/etc/svxlink/svxlink.conf*):

```
[RepeaterLogic]
TYPE=Repeater
RX=QuantarRx
TX=MultiTx
MODULES=ModuleEchoLink,ModuleMetarInfo,ModulePropagationMonitor,ModuleSelCallEnc,ModuleTrafficInfo
....
[MultiTx]
TYPE=Multi
TRANSMITTERS=RepeaterTx,StreamingTx
[RepeaterTx]
TYPE=Local
AUDIO_DEV=alsa:hw0
AUDIO_CHANNEL=0
PTT_PORT=/dev/ttyS0
PTT_PIN=DTRRTS
TX_DELAY=0
PREEMPHASIS=0
DTMF_TONE_LENGTH=100
DTMF_TONE_SPACING=50
DTMF_TONE_AMP=-18
[StreamingTx]
TYPE=Local
AUDIO_DEV=alsa:dmix1
AUDIO_CHANNEL=0
PTT_PORT=NONE
PTT_PIN=DTRRTS
TX_DELAY=0
PREEMPHASIS=0
DTMF_TONE_LENGTH=100
DTMF_TONE_SPACING=50
DTMF_TONE_AMP=-18
```

Jetzt alle Applikationen starten und die Funktion prüfen.

Viel Erfolg wünscht Adi, DL1HRC

Internetstreaming ab Linux Kernel Version 2.6.37

Ab der Linux Kernel Version 2.6.37 kann auf eine 2. Soundkarte verzichtet werden,
da ab dieser Kernel-Version das Modul “snd-aloop” was ein loopback-Device erzeugt bereits mitgeliefert wird.

Die folgende Beschreibung bezieht sich auf OpenSuSE12.1 sie sollte aber auch leicht auf andere Distributionen übertragbar sein.

Voraussetzungen:
* einen Streaming-Server (Icecast2 oder Shoutcast)
* einen Streaming-Client (Ices2 oder SC_Trans)
* Linux Kernel ab Version 2.6.37 auf dem SvxLink-PC (z.B. OpenSuSE 12.1)

Zunächst müssen Icecast2 und Ices2 oder bei Verwendung von Shoutcast der Shoutcast-Server und SC_Trans installiert und nach Wunsch konfiguriert werden, im Internet gibt es hierfür mehr als genügend Installations- und Konfigurationsanleitungen.

Der Streaming-Server muss nicht zwingend auf dem selben PC installiert werden auf dem auch SvxLink läuft, hier bietet sich schon allein wegen der besseren Netzwerkanbindung ans Internet ein root-Server an, soweit dieser zur Verfügung steht.

Ices2 oder auch SC_Trans werden entweder auf dem PC auf dem SvxLink läuft oder wie bei mir, auf dem PC auf dem remotetrx läuft (um den SvxLink-PC von der CPU-Last zu befreien) installiert.

Dann müssen wir sicherstellen das die Module “snd-aloop” und “snd-pcm-oss”, dieses wird für die Verbindung zum Streaming-Client benötigt, beim starten des Systems auch geladen werden.

[/etc/sysconfig/kernel]:

```
MODULES_LOADED_ON_BOOT="snd-aloop snd-pcm-oss"
```

Nach dem Neustart des Systems kann nun mit dem Befehl “aplay -l” überprüft werden ob das Modul “snd-aloop” auch geladen wurde.

```
remote-trx1:~ # aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: rev40 [VIA 82C686A/B rev40], device 0: VIA 82C686A/B rev40 [VIA 82C686A/B rev40]
Subdevices: 0/1
Subdevice #0: subdevice #0
card 1: Loopback [Loopback], device 0: Loopback PCM [Loopback PCM]
Subdevices: 8/8
Subdevice #0: subdevice #0
Subdevice #1: subdevice #1
Subdevice #2: subdevice #2
Subdevice #3: subdevice #3
Subdevice #4: subdevice #4
Subdevice #5: subdevice #5
Subdevice #6: subdevice #6
Subdevice #7: subdevice #7
card 1: Loopback [Loopback], device 1: Loopback PCM [Loopback PCM]
Subdevices: 8/8
Subdevice #0: subdevice #0
Subdevice #1: subdevice #1
Subdevice #2: subdevice #2
Subdevice #3: subdevice #3
```

Subdevice #4: subdevice #4

Subdevice #5: subdevice #5

Subdevice #6: subdevice #6

Subdevice #7: subdevice #7

Wenn soweit alles funktioniert und auch das Modul “snd-pcm-oss” ordnungsgemäß geladen wurde dann kann nun “*/dev/dsp1*” als Capturedevice für den Streaming-Client verwendet werden.

[sc_trans.conf]:

...

capture=1

capturedevice=/dev/dsp1

captureinput=Capture

capturesamplerate=16000

capturechannels=1

...

[ices2.xml]:

...

< stream >

< metadata >

< name >DB0DIN< /name>

< genre >Hamradio Network< /genre >

< description >DB0DIN Livestream< /description >

< url >http://db0din.de< /url >

< /metadata >

< input >

< module >oss< /module >

< param name="rate">16000< /param>

< param name="channels">1< /param>

< param name="device">/dev/dsp1< /param>

< param name="buffer-time">500< /param>

< param name="metadata">1< /param>

< param name="metadatafilename">/tmp/info.txt< /param>

< /input>

...

SvxLink oder der remotetrx wird noch als Multi-Tx konfiguriert

[/etc/svxlink/svxlink.conf oder **/etc/svxlink/remotetrx.conf]:**

[RepeaterLogic]

TYPE=Repeater

RX=QuantarRx

TX=MultiTx

MODULES=ModuleEchoLink,ModuleMetarInfo,ModulePropagationMonitor,ModuleSelCallEnc,ModuleTrafficInfo

....

[MultiTx]

TYPE=Multi

TRANSMITTERS=RepeaterTx,StreamingTx

[RepeaterTx]

TYPE=Local

AUDIO_DEV=alsa:plughw:0

AUDIO_CHANNEL=0

PTT_PORT=/dev/ttyS0

PTT_PIN=DTRRTS

TX_DELAY=0

PREEMPHASIS=0

DTMF_TONE_LENGTH=100

DTMF_TONE_SPACING=50

DTMF_TONE_AMP=-18

[StreamingTx]

TYPE=Local

AUDIO_DEV=alsa:plughw:1,1 #hier wird nun das 1. capture-device des snd-aloop ausgewählt

AUDIO_CHANNEL=0

PTT_PORT=NONE

PTT_PIN=DTRRTS

TX_DELAY=0

PREEMPHASIS=0

Jetzt alle Applikationen starten und die Funktion prüfen.

Viel Erfolg wünscht do6ep, Mathias

Tcl-Routinen

Die TCL-Skripte im Verzeichnis */usr/share/svxlink/events.d/* stellen sozusagen die Präsentationsebene von SvxLink dar. Hier wird festgelegt auf welche Art und Weise beim Eintreten von bestimmten Ereignissen die Ausgaben erfolgen, ob z.B. vor oder nach einer Ansage noch ein Ton gesendet wird, wenn ja in welcher Höhe und Länge…usw.

Ein paar Beispiele

Ereignis	Funktion im Tcl-Skript
Öffnen oder Schliessen der Rauschsperre	Logic.tcl -> squelch_open{}
Schließen der Rauschsperre und Ablauf des RGR-Timers	Logic.tcl -> send_rgr_sound{}
Empfang des DTMF-digits "*"	Logic.tcl -> manual_identification{}
Sender ein oder aus	Logic.tcl -> transmit{}

Beispiel für Änderungen am Roger-Piep

Der für den Roger-Piep zuständige Abschnitt in der */usr/share/svxlink/events.d/Logic.tcl*ist die Prozedur *send_rgr_sound{}*:

```
#
# Executed when the squelch just have closed and the RGR_SOUND_DELAY timer has
# expired.
#
proc send_rgr_sound {} {
 variable sql_rx_id;
 playTone 440 80 50;
 playSilence 50;
  for {set i 0} {$i < $sql_rx_id} {incr i 1} {
  playTone 880 50 50;
  playSilence 50;
 }
 playSilence 100;
}
```

playTone 440 80 50; erzeugt einen Ton von 440 Hertz mit eine Amplitude von 80 und einer Länge von 50 Millisekunden. playSilence 50; liefert eine Pause von 50 ms. Der darauf folgende Abschnitt

```
for {set i 0} {$i < $sql_rx_id} {incr i 1} {
 playTone 880 50 50;
 playSilence 50;
}
```

ist nur interessant, wenn Sie mehrere remote-trx betreiben. Anhand der Anzahl der nachfolgenden Töne läßt sich ermitteln, über welchen Rx die vorhergehende Sendung aufgenommen wurde.

In der Logic.tcl gibt es ca. 38 Routinen, welche bei unterschiedlichsten Ereignissen ausgeführt werden, so dass sich sicher für jeden nur erdenklichen Fall eine Konfiguration für ein bestimmtes Ereignis realisieren läßt. Achte Sie aber bei Anpassungen darauf, dass diese in userspezifischen tcl-Dateien passieren und NICHT in den mitgelieferten tcl-Routinen. Weitere Informationen finden Sie unter "locale-Konzept": http://svxlink.de/?page_id=903

Parameter zur Laufzeit ändern

Es gibt eine relativ einfache Möglichkeit, TCL-Variablen zur Laufzeit von SvxLink zu ändern und diese innerhalb der TCL-Funktionen auszuwerten. Ein kleines Beispiel soll das Prinzip verdeutlichen.

Gefordert sei: Beim Connect einer Station via EchoLink soll an diese Station eine konfigurierbare Ansage erfolgen, die z.B. darüber informiert, dass z.B. DL1ABC nicht an der Station ist.

Erstellen Sie eine Datei (z.B. */tmp/user_params.tcl*), die verschiedene Variablendefinitionen enthält:

```
set away 1;

set ansage abwesenheitsinformation;
```

Erstellen Sie eine Datei (*/usr/share/svxlink/events.d/locale/el.tcl*) nach dem locale-Prinzip.

```
namespace eval EchoLink {

  proc remote_greeting {} {

    variable away;

    variable ansage;

    source "/tmp/user_params.tcl";

    playSilence 1500;

    if {$away == 1} {

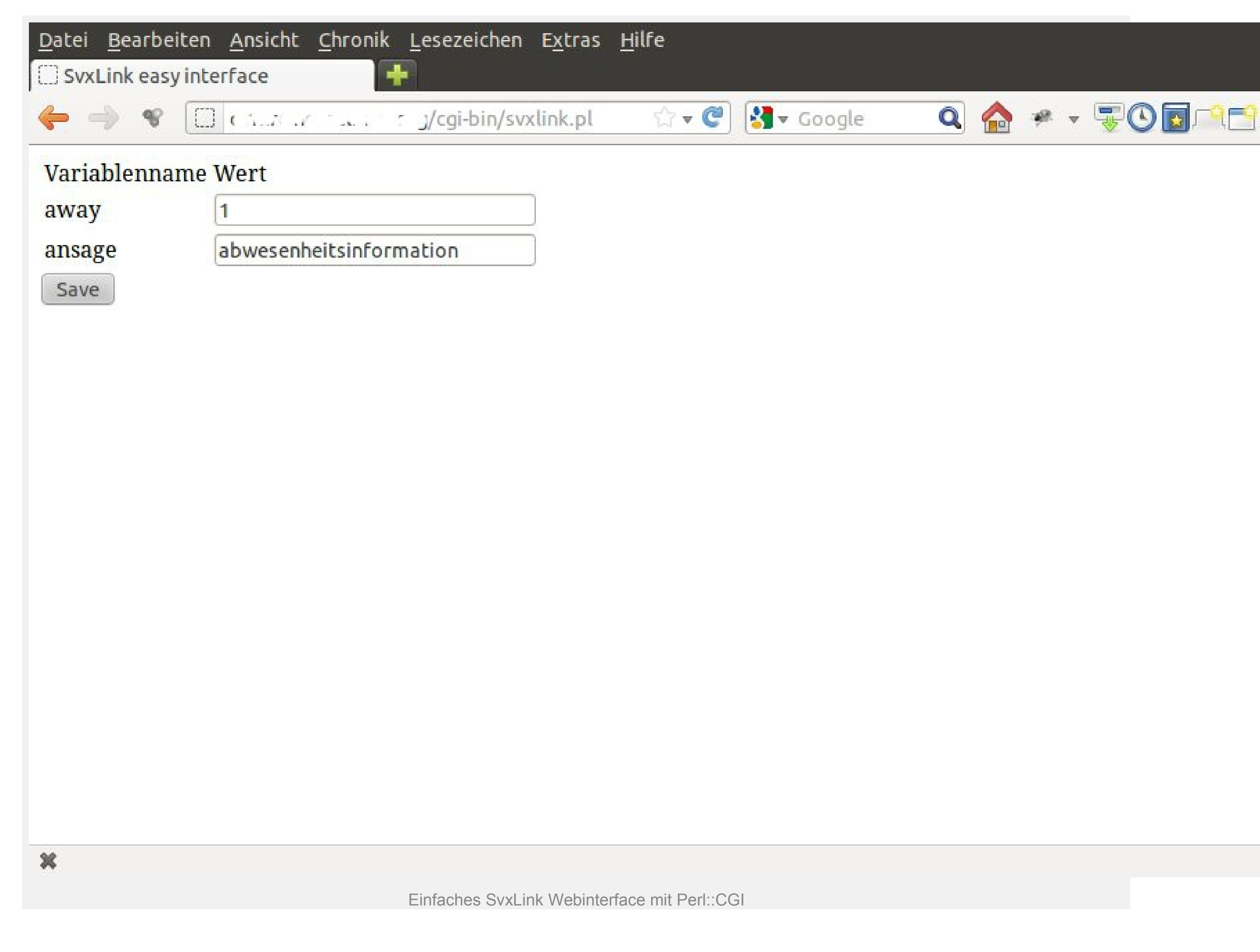
    playMsg $ansage;

  }

 }

}
```

Um die Parameter zu ändern, kann man einfach das file */tmp/user_params.tcl* bearbeiten oder mein kleines Webinterface installieren (apache2, perl, perl::CGI sind notwendig). Im Beispiel wird für die eingehende EchoLink-Verbindung die Audiodatei*abwesenheitsinformation.wav* abgespielt, wenn away=1 gesetzt ist.



Einfaches SvxLink Webinterface mit Perl::CGI

Auf diese Weise könnte man z.B. auch Wetterdaten aus einem File einlesen, welches periodisch mit den aktuellen Daten einer angeschlossenen Wetterstation gefüttert wird. Das File müsste dann in Form eines einfachen TCL-Files erstellt werden, d.h. in folgender Form (*/tmp/wetterdaten.tcl*):

```
set wind 100;

set temperatur 33;

set taupunkt 22;
```

...usw

Zur vollen Stunde könnte dieses File gesourced werden und neben der aktuellen Uhrzeit kann die aktuelle Temperatur mit ausgegeben werden, hier das Prinzip:

```
...

source "/tmp/wetterdaten.tcl";

playMsg "die_temperatur_betraegt";

playNumber $temperatur;

playMsg "grad_celsius";

...
```

Persönliche Begrüßung eingehender EchoLink-Stationen

Seit dem Trunk, Rel. 2223 (ab 18.08.2012) ist es möglich einer eingehenden EchoLink-Station eine persönliche Begrüßung zu übermitteln (rufzeichenspezifisch). Hierzu ist unter Beachtung des locale-Konzeptes eine kleine tcl-Datei im Verzeichnis */usr/share/svxlink/events.d/local* zu erstellen, Dateiname z.B. *user_greeting.tcl*:

```
# Ergänzung zur individuellen Begrüssung

# by Michael Hartje, DK5HH



namespace eval EchoLink {



#

# Executed when an incoming connection is accepted

#

proc remote_greeting {callsign} {

  global langdir; # aus der Hauptebene in events.tcl

  set callsign [string toupper $callsign]; # nur zur Sicherheit

  set callsign [string map {- " "} $callsign; # Trennzeichen verwenden

  set found [scan $callsign "%s%s" call suffix];

  if {$found > 0 && [file exists "$langdir/EchoLink/greeting-$call.wav"]} {

    playSilence 500;

    playFile "$langdir/EchoLink/greeting-$call.wav";

  } else {

    playSilence 1000;

    playMsg "greeting";

  }

}



}

# End Namespace
```

Die Funktionsweise ist recht einfach, im Verzeichnis */usr/share/svxlink/sounds/de_DE/EchoLink* sind für eine deutsche Sprachumgebung die personifizierten Begrüßungsdateien im Format *greeting-CALL.wav* abzulegen, wobei CALL durch das betreffende Rufzeichen ersetzt wird. Bei einer eingehenden EchoLink-Verbindung (z.B.

von DK5HH) wird im Soundverzeichnis nach einer Datei *greeting-DK5HH.wav* gesucht, ist diese vorhanden wird sie abgespielt.

Trennung EchoLink-Verbindung verhindern

Es gibt leider eine ganze Reihe von Leuten, deren Horizont arg beschränkt ist, die dem QSO-Thema nicht folgen können, den OMs durch die Trennung der EchoLink-Verbindung permanent das QSO zerstören oder für Rundsprüche zusammengeschaltete Relaisstationen trennen. Um zu verhindern, dass eine EchoLink-Verbindung ständig durch diese Störer getrennt wird, kann man einen der folgenden Wege gehen.

1) Disconnect wird durch "geheimes Kommando" verhindert

* Erzeugen Sie eine Datei z.B. *no-disconnect.tcl* mit dem folgenden Inhalt:

```
# das Überschreiben der Prozedur mit folgendem Inhalt verhindert das

# einfache Trennen einer EchoLink-Verbindung durch "#"

# Ersetzen Sie die Werte für

# "XX" mit dem Kommando für die Aktivierung der Disconnect-Sperre

# "YY" mit dem Kommando für die Deaktivierung der Disconnect-Sperre

# für den Zeitraum der Aktivierung nimmt svxlink kein "leeres" DTMF-Kommando

# entgegen (z.B. Rundspruch)



namespace eval Logic {



  # variable for Logic.tcl (muss im namespace Logic definiert werden, Nachtrag, 5.3.13, M. H.)

  variable ignore_dtmf 0;

  proc dtmf_cmd_received {cmd} {

     puts "DTMF command received: $cmd";

     variable ignore_dtmf;

     if {$cmd == "XX"} {

         set ignore_dtmf 1;

         return 1;

     }

     if {$ignore_dtmf} {

      if {$cmd == "YY"} {
```

```
        set ignore_dtmf 0;
        return 1;
      } elseif {$cmd == ""} {
        return 1;
      }
    }
  return 0;
 }
}
# end of namespace Logic
```

2) Disconnect ist erst nach xx Sekunden möglich und nur wenn von der EchoLink-Station kein Audiostream mehr zu empfangen ist~~, benötigt eine Erweiterung im Modul EchoLink/ModuleEchoLink::inIsReceiving~~():
* Erzeugen Sie eine Datei z.B. *no-disconnect.tcl* mit dem folgenden Inhalt:

```
namespace eval Logic {
  proc dtmf_cmd_received {cmd} {
    global last_el_stream;
    global is_el_rx;
    variable now;
    variable ignore_dtmf;
    variable diff;
    set cs $EchoLink::num_connected_stations;
    set now [clock seconds];


    # Trennung erst nach 35 Sekunden oder durch geheimes Kommando "78#"
    set diff [expr {$last_el_stream + 35 - $now}];
    if {$cs > 0} {
      if {(($diff < 0) && !$is_el_rx) || $cmd == "78"} {
         puts "disconnect OK";
         return 0;
      }
```

```
        return 1;
        } else {
          return 0;
        }
    }
}
# end of namespace Logic


namespace eval EchoLink {
   proc is_receiving {isrec} {
     global last_el_stream;
     global is_el_rx;
     set is_el_rx $isrec;
     if {$isrec == 0} {
       set last_el_stream [clock seconds];
     }
    puts "is_receiving $isrec $is_el_rx $last_el_stream";
  }
}
# end of namespace EchoLink
```

2) Speichern Sie diese Datei(en) in ein entsprechendes locale-Verzeichnis (siehe Locale-Konzept) und starten Sie SvxLink neu

Unterschiedliche Roger-Beep's

Für die Identifizierung einer EchoLink-Verbindung und HF-Verbindung kann man unterschiedliche Roger-Beeps ausgeben. Um das zu realisieren muss man die Prozeduren*Logic::squelch_open* und *Logic::send_rgr_sound* überschreiben. Hier die Kurzanleitung für eine deutsche Sprach-Umgebung:

1) eine Datei z.B. *roger-beep.tcl* erstellen mit folgendem Inhalt:

```
#
# zwei verschiedene Roger-beeps:
# hoher Ton (540Hz) - EchoLink
```

```
# tieferer Ton (440Hz) - HF-Verbindung
#
namespace eval Logic {

variable is_rf 0;

proc squelch_open {rx_id is_open} {
  variable sql_rx_id;
  variable is_rf;
  set sql_rx_id $rx_id;
  if {!$is_open} {
    playSilence 100;
    playTone 440 70 50; # Ton mit 440Hertz ausgeben
    set is_rf 1;
  }
}

proc send_rgr_sound {} {
 variable is_rf;
 if {!$is_rf} {
   playTone 540 70 50; # Ton mit 540Hertz ausgeben
 }
 set is_rf 0;
 }
}
# end of namespace
```

2) diese Datei in das Verzeichnis */usr/share/svxlink/events.d/local/* speichern
3) SvxLink neu starten

Informationsansagen zeitlich gesteuert mit cron

Vorbereitete Informationsansagen kann man relativ einfach kurz vor dem Abfallen des Relais aussenden — also zu einem Zeitpunkt, an dem noch jemand zuhört, aber eigentlich niemand mehr sprechen möchte. Zu diesem Zeitpunkt gibt es zumeist mindestens einen Zuhörer, der vorher das Relais vielleicht nur aufgetastet hatte.

Die Informationsansagen sollten nicht mehr als 20 s lang sein. Dazu werden die vorbereiteten WAV-Dateien in einem Verzeichnis unterhalb von
/usr/share/svxlink/sounds/de_DE abgelegt, z.B. in *LocalAudio*. Die Dateien sollten alle mit “*Idle-*” beginnen, also zB. *Idle-00-OV-Abend-Einladung.wav*
Weitere Ansagen zur Benutzung des Relais, den vereinbarten Kurzwahlen, helfen Durchreisenden oder anderen noch nicht umfassend Informierten, sich schneller zurecht zu finden.

Mit der folgenden Prozedur unter */usr/share/svxlink/sounds/de_DE/events.d/local*können die vorbereiteten Sounddateien jedes Mal durchgesehen und dann abgespielt werden.

```
#

# Abspielen von Info-wav bei Idle des Relais

#
```

```
# Entwurf DK5HH

#


namespace eval RepeaterLogic {


 #Counter which Idle-sound shall be the next

 variable idle_nr 0;

 variable last_idlesound 100000000;


 # minimum time between 2 idle sounds in seconds

 variable min_time_between_idlesound 300;


 proc repeater_idle {} {

  set filelist [get_idle_files];

  set idle_max [llength $filelist];

  set filelist [lsort -ascii $filelist];

  variable idle_nr;

  variable last_idlesound;

  variable min_time_between_idlesound;

  set now [clock seconds];

  # more than 300 s since last idlesound

  if {$now - $last_idlesound > $min_time_between_idlesound } {

    set last_idlesound $now;

    if { $idle_nr >= $idle_max } {

     set idle_nr 0;

    }

    set psound [lindex $filelist $idle_nr]

    set idle_nr [expr {$idle_nr + 1} ];
```

```
    puts "playing Idle-Sound $psound. $idle_nr\/$idle_max";

    playMsg "LocalAudio" $psound;

    playSilence 800;

  }

 }

 proc get_idle_files {} {

  variable dir "/usr/share/svxlink/sounds/de_DE/LocalAudio";

  variable ext "wav";

  variable files "";

  foreach file [glob -nocomplain -directory $dir Idle-*.wav] {

    set foundfile [file tail [file rootname $file]];

    lappend files $foundfile;

  }

  return $files;

 }



}

# end Namespace RepeaterLogic
```

Durch *cron* kann man die Ansagen durch Umbenennen zu einem bestimmten Zeitpunkt aktivieren oder deaktivieren — also zum OV-Abend eine Woche vorher die Ansage aktivieren und während des OV-Abends wieder deaktivieren. Also: z.B. aus *Idle-….* ein*idle-….* machen. — Dann kann die Ansage "nächste Woche Dienstag ist OV-Abend, willst Du nicht auch kommen?" einladend wirken.
Ein beliebiger User der Gruppe *svxlink* kann einen Eintrag mit *crontab -e* anlegen. Dabei wird das folgende kleine Shell-Skript *idle-activate.sh* verwendet:

```
#! /bin/bash

# wird mit cron aufgerufen zum Aktivieren / Dekativieren

# von Idle-Sounds

DIR=/usr/share/svxlink/sounds/de_DE/LocalAudio/

cd $DIR

ACTIVE=$1
```

```
if [ -z $ACTIVE ]
  then
    echo "usage: Parameter EIN|AUS \"Datei-pattern\""
  fi
#U_ACT=tr '[:lower:]' '[:upper:]' $ACTIVE
if [ $ACTIVE = "EIN" ]
  then
    for renfile in $(find ./ -name "$2" )
      do
        re1=$(basename $renfile)
        re2=${re1/idl/Idl}
        mv $re1  $re2
      done
fi
if [ $ACTIVE = "AUS" ]
  then
    for renfile in $(find ./ -name "$2" )
      do
        re1=$(basename $renfile)
        re2=${re1/Idl/idl}
        mv $re1  $re2
      done
fi
```

das kleine Shell-Hilfsskript *idle-activate.sh* legen wir im **$HOME**-Verzeichnis dieses Benutzers (zB dk5hh) ab und machen es mit

```
chmod 775 idle-activate.sh
```

ausführbar.

Selektivruf bei eingehender EchoLink-Verbindung auslösen

In vielen Fällen möchte man den oft langatmigen und weniger interessanten Relais-Gesprächen nicht ausgesetzt sein. Um aber trotzdem für <<wichtige>> EchoLink-Stationen erreichbar zu sein, kann man bei kommerziellen Geräte wie z.B. dem Motorola GM1200 eine Selektivruffunktion aktivieren (ZVEI1, EIA,…). Für den tcl-Namespace ist dann eine kleine Anpassungen erforderlich.

1) Aktivieren Sie den SelCall-Encoder in der SvxLink-Konfiguration, siehe:http://svxlink.de/?page_id=187
2) Erstellen Sie unter Beachtung des locale-Konzeptes eine tcl-Datei, z.B. *sel5.tcl* mit folgendem Inhalt:

```
#
# This routine sends out a Sel5call over Rf
# when a VIP-EchoLink station was connecting
# the own node
#
namespace eval EchoLink {

  proc remote_connected {call} {

    SelCall::setMode "ZVEI1"; # define the SelCall-mode

    if {$call == "DL1ABC" || $call == "SM0SVX-L"} {

      SelCall::play "01231"; # the own station with call
                             # "01231" configured
    }

    playSilence 500;
    playMsg "connected";
    spellEchoLinkCallsign $call;
  }
}
```

3) Starten Sie SvxLink neu und warten Sie auf die nächste eingehende EchoLinkverbindung von “DL1ABC” oder “SM0SVX-L”, bitte an die eigenen Erfordernisse anpassen. Stellt eine der konfigurierte Partner-Stationen eine

Verbindung zum eigenen EchoLink-Node her, so wird ein Selektivruf über HF ausgesendet, in diesem Falle die Tonfolge zum Ruf “01231″.

Beeps bei EchoLink-Verbindung unterdrücken (z.B. I-Phone)

Speziell bei EchoLinkverbindungen über IPhones kann es zu Problemen in der Form kommen, dass während der Übertragung von dieser Station viele Roger-Pieps ausgelöst werden weil eine kurze Unterbrechung des Audiostreams als Ende der Übertragung erkannt werden.

Um das abzustellen bitte folgende Änderungen durchführen:

1) Im Verzeichnis */usr/share/svxlink/events.d* solte es ein Verzeichnis “local” geben, falls nicht bitte anlegen.
2) Erstellen sie in diesem Verzeichnis eine Datei, z.B. EchoLink-user.tcl mit dem folgenden Inhalt:

```
namespace eval EchoLink {



#

# Executed when a transmission from an EchoLink station is starting

# or stopping

#

proc is_receiving {rx} {

  #if {$rx == 0} {

  #  playTone 1000 100 100;

  #}

}



# end of namespace

}



#

# This file has not been truncated

#
```

Zeitlich gesteuerte Ausgaben generieren

Viele OM’s möchten gern per automatischer Relaisansage beispielsweise auf den nächsten OV-Abend hinweisen. Um so eine Ausgabe zu realisieren gibt es viele Ansätze, im Folgenden sollen drei davon vorgestellt werden. Alle basieren auf dem SvxLink-eigenen TCL-Framework und darin enthaltenen Zeitfunktionen.

Für den weiteren Verlauf wird davon ausgegangen, dass eine Ansage-Datei mit Namen “OV-Abend.wav” bereits erstellt und ins Verzeichnis*/usr/share/svxlink/events.d/de_DE/Core* kopiert wurde (oder unter ../en_US/… je nach Locale-Umgebung).

1) Nutzung der every_minute{}-Prozedur

Im TCL-Framework gibt es eine Prozedur, die zu jeder vollen Minute aufgerufen wird, die every_minute{}-Prozedur. Diese kann man zur Ausgabe heranziehen, allerdings indirekt über addTimerTickSubscriber{}!
Erstellen Sie nach dem locale-Konzept eine Datei *meine_ansage.tcl* mit folgendem Inhalt:

```
# /usr/share/svxlink/events.d/local/meine_ansagen.tcl

namespace eval Logic {

 Logic::addTimerTickSubscriber relaisansage;



 proc relaisansage {} {

   puts "Ausgabe Relaisansage?";

   source "/home/svxlink/relaisansage.tcl";

  }

}
```

Erstellen Sie eine Datei *relaisansage.tcl* im */home/svxlink*-Verzeichnis mit folgendem Inhalt:

```
# /home/svxlink/relaisansage.tcl

# aktuelle Uhrzeit ermitteln

set now [clock seconds];

set hour [clock format $now -format "%H"];

set minute [clock format $now -format "%M"];



# eigene Ausgabezeiten definieren

set ansage_Minute 05;

set ansage_Stunde 09;



puts "Es ist $hour:$minute, Ausgabe um $ansage_Stunde:$ansage_Minute";
```

```
# prüfen ob Ausgabezeit erreicht ist

if {$hour == $ansage_Stunde && $minute == $ansage_Minute} {

  puts "Ausgabe Relaisansage";

  playMsg "Core" "OV-Abend";

}
```

Alle Zeilen mit **puts** beginnend dienen nur der Kontrollausgabe und können selbstverständlich weggelassen bzw. auskommentiert werden. Wichtig ist, die gewünschte Ausgabezeit mit führenden Nullen bei Werten kleiner 10 einzutragen, also 09 statt 9!
Wird eine periodisch über den Tag verteilte Ausgabe gewünscht so kann man hier die Modulo-Funktion zielführend einsetzen. Ändern Sie hierzu die Datei*/home/svxlink/relaisansage.tcl* wie folgt:

```
# /home/svxlink/relaisansage.tcl

# aktuelle Uhrzeit ermitteln

set now [clock seconds];

set hour [clock format $now -format "%H"];

set minute [clock format $now -format "%M"];



# eigene Ausgabezeiten definieren

set interval 6;

set ansage_Minute 22;



# hier prüfen ob Ausgabekriterium erreicht ist

if {($hour % $interval) == 0 && $minute == $ansage_Minute} {

  puts "Ausgabe Relaisansage";

  playMsg "Core" "OV-Abend";

}
```

In diesem Beispiel würde die Ausgabe zur 22. Minute einer ohne Teilungsrest durch 6 teilbaren Stunde erfolgen, also um 0:22, 6:22, 12:22 und 18:22.

2) Nutzung der periodischen Relais-Identifikationfunktion
Erstellen Sie nach dem locale-Konzept eine Datei “meine_ansagen.tcl” mit nachfolgendem Inhalt. Achten Sie darauf, dass der Inhalt der Funktion Logic::send_long_ident{} dem Standard nach einer Neuinstallation entspricht. Falls Sie also diese Funktion schon geändert haben (z.B. per locale-Konzept in einer anderen tcl-Datei), so sollten Sie die gewünschten Änderungen dort durchführen!

```
# /usr/share/svxlink/events.d/local/meine_ansagen.tcl
```

```
namespace eval Logic {


#

# Executed when a long identification (e.g. hourly) should be sent

#   hour    - The hour on which this identification occur

#   minute  - The hour on which this identification occur

#

proc send_long_ident {hour minute} {

  global mycall;

  global loaded_modules;

  global active_module;

  variable CFG_TYPE;


  spellWord $mycall;

  if {$CFG_TYPE == "Repeater"} {

    playMsg "Core" "repeater";

  }

  playSilence 500;


  playMsg "Core" "the_time_is";

  playSilence 100;

  playTime $hour $minute;

  playSilence 500;


  # Call the "status_report" function in all modules if no module is active

  if {$active_module == ""} {

    foreach module [split $loaded_modules " "] {

      set func "::";
```

```
      append func $module "::status_report";

      if {"[info procs $func]" ne ""} {

        $func;

      }

    }

  }

  playSilence 250;

  source "/home/svxlink/relaisansage.tcl";

  playSilence 250;

}

}
```

Erstellen Sie dann wieder eine Datei *relaisansage.tcl* im */home/svxlink*-Verzeichnis. Der Inhalt kann jetzt etwas einfacher sein, muß aber nicht. Es ist jetzt allerdings wichtig geworden, wie oft Sie den LONG_IDENT_INTERVAL ausgeben lassen. Hat man LONG_IDENT_INTERVAL=120 konfiguriert, so würde die Datei relaisansage.tcl nur jede 2h aufgerufen werden:

```
# /home/svxlink/relaisansage.tcl

puts "Ausgabe Relaisansage";

playMsg "Core" "OV-Abend";
```

Natürlich kann man das mit dem Vergleich einer vordefinierten Stunde kombinieren wie unter Punkt 1) beschrieben. Hier muß man einfach ein bisschen probieren. Bitte vorher überlegen, ob das Kriterium überhaupt **true** werden kann.
Ein grosser Vorteil dieser Varianten ist, dass Änderungen in o.g. Datei sofortige Auswirkungen haben. Man kann diese Ausgaben quasi on-thy-fly herausnehmen, indem man in der */home/svxlink/relaisansage.tcl* einfach das

```
playMsg "Core" "OV-Abend";
```

auskommentiert. Ein Neustart ist nicht erforderlich. Bitte diese Datei nicht löschen!

3) Ansagen vor dem “Abfallen” einer Relaisstation
Diese Variante funktioniert nur bei einer RepeaterLogic. Durch eine Änderung der Prozedur *RepeaterLogic::repeater_down{}* (locale-Konzept!) kann man auf einfache Weise realisieren, dass die gewünschte Ausgabe z.B. bei jedem 20. Relaisschließen erfolgen soll. Erstellen Sie eine Datei *meine_ansage.tcl* nach dem locale-Konzept und prüfen Sie, ob Sie eventuell eine Änderung dieser Prozedur schon irgendwo*RepeaterLogic::repeater_down{}* vorgenommen haben.

```
# /usr/share/svxlink/events.d/local/meine_ansagen.tcl

namespace eval RepeaterLogic {

#

# Executed when the repeater is deactivated
```

```
#   reason  - The reason why the repeater was deactivated
#             IDLE         - The idle timeout occured
#             SQL_FLAP_SUP - Closed due to interference
#
proc repeater_down {reason} {
  global mycall;
  variable repeater_is_up;
  variable count;
  incr count;

  set repeater_is_up 0;

  if {$reason == "SQL_FLAP_SUP"} {
    playSilence 500;
    playMsg "Core" "interference";
    playSilence 500;
    return;
  }

  if {$count > 19} {
    source "/home/svxlink/relaisansage.tcl";
    set count 0;
  }

  set now [clock seconds];
  if {$now-$Logic::prev_ident < $Logic::min_time_between_ident} {
    playTone 400 900 50
    playSilence 100
```

```
    playTone 360 900 50

    playSilence 500

    return;

  }

  set Logic::prev_ident $now;



  spellWord $mycall;

  playMsg "Core" "repeater";

  playSilence 250;

  #playMsg "../extra-sounds" "shutdown";

}



# end of namespace

}
```

Erzeugen Sie wieder die bekannte Datei "relaisansage.tcl" im /home/svxlink-Verzeichnis:

```
# /home/svxlink/relaisansage.tcl

puts "Ausgabe Relaisansage";

playMsg "Core" "OV-Abend";
```

SvxLink für 16k-Sounds selber kompilieren (obsolete)

ACHTUNG: Aktuelle SvxLink-Trunks und -Releases sind bereits für 16k-Sounds konfiguriert! Sie müssen nichts mehr umkonfigurieren!
SvxLink wurde bis 2013 standardmäßig als fertiges Paket ausgeliefert, welches nur 8k-(wav|raw)-Dateien unterstützt. Diese klingen sagen wir mal "etwas digital". Eine wesentlich bessere Qualität kann man von 16k-Sounds erwarten. Hier ist noch etwas Handarbeit gefragt, man muß die Quellen noch selber kompilieren.

1) Laden Sie sich die Quellen per svn herunter, entweder das offizielle Release:

```
svn co https://svxlink.svn.sourceforge.net/svnroot/svxlink svxlink-release
```

oder für die mutigen SvxLink-Nutzer die Trunk-Version

```
svn co https://svxlink.svn.sf.net/svnroot/svxlink/trunk svxlink-trunk
```

2) wechseln Sie in das entsprechende src-Verzeichnis und ändern Sie in der Datei*makefile.cfg* den Eintrag wie folgt ab:

```
CXXFLAGS += -DINTERNAL_SAMPLE_RATE=8000
```

in

```
CXXFLAGS += -DINTERNAL_SAMPLE_RATE=16000
```

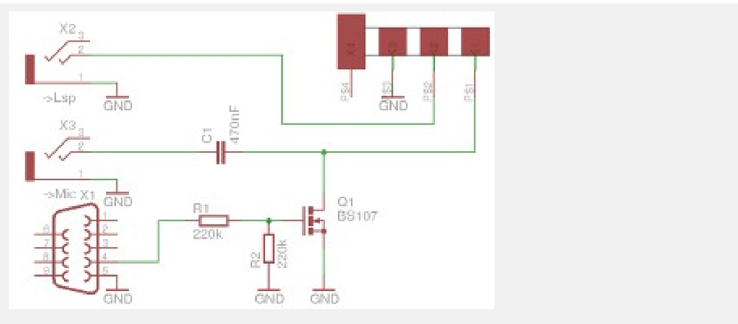
3) führen Sie die Schritte *make* sowie *make install* aus.
Ab jetzt können Sie die 16k-Sounds nutzen.

XTS3000 an SvxLink

Die XTS3000 von Motorola sind Handfunkgeräte, die sowohl analogen als auch digitalen (Apco25) Betrieb erlauben. Es liegt also nahe, einen einfachen Adapter zu bauen, der SvxLink mit der digitalen Apco25-Welt verbindet.

Einen Audio-Adapter für die XTS-Serie gibt es für wenig Geld bei einem einschlägigen Auktionshaus, mit etwas probieren habe ich auch die Anschlussbelegung herausgefunden. Man benötigt einen 3.5mm Klinkenstecker mit insgesamt 4(!) Kontakten wobei der Außenring (Masse) nicht genutzt wird. Belegung der Kontakte am 4poligen Klinkenstecker (Ohne Garantie!):

1. ext. Lautsprecher (über Kondensator) / PTT
2. Mikrophon
3. Masse
4. (Außenring) N.C.




XTS3000-SvxLink Adapter


XTS3000 Audioadapter

Leider kann SvxLink nicht per Apco25-Datenstrom in irgendeiner Weise gesteuert werden, dazu müsste man diesen dekodieren und auswerten (per RS232, z.B.). Es wird also lediglich eine Audioverbindung geschaltet.

Die Konfiguration in SvxLink entspricht der einer MicSpkrLogic mit PTT. Als Soundkarte dient eine einfache USB-Soundkarte, die für knapp 5€ beim bereits erwähnten Auktionshaus erworben werden kann. Für diesen Anwendungsfall ist das vollkommen ausreichend. Nachfolgend eine Beispielkonfiguration:

```
[ApcoLogic]
TYPE=Simplex
RX=Apco-Micro
TX=Apco-Spekr
MODULES=ModuleHelp
DEFAULT_LANG=de_DE
```

CALLSIGN=APCO

EVENT_HANDLER=/usr/share/svxlink/events.tcl

RGR_SOUND_DELAY=10

NO_REPEAT=1

SHORT_IDENT_INTERVAL=60

LONG_IDENT_INTERVAL=60

MUTE_RX_ON_TX=1

[Apco-Micro]

TYPE=Local

AUDIO_DEV=alsa:plughw:2

AUDIO_CHANNEL=0

SQL_DET=VOX

SQL_START_DELAY=0

SQL_DELAY=0

SQL_HANGTIME=1700

SQL_TIMEOUT=600

VOX_FILTER_DEPTH=20

VOX_THRESH=1000

SERIAL_PORT=/dev/ttyUSB1

SERIAL_PIN=DCD:SET

PEAK_METER=0

DTMF_DEC_TYPE=INTERNAL

DTMF_MUTING=1

DTMF_HANGTIME=100

DTMF_SERIAL=NONE

[Apco-Spekr]

TYPE=Local

AUDIO_DEV=alsa:plughw:2

AUDIO_CHANNEL=1

PTT_PORT=/dev/ttyUSB1

PTT_PIN=DTR

TX_DELAY=0

TIMEOUT=1200



Anschlus eines XTS3000 an SvxLink (DB0WSF-R)

Compileroptionen für ARM, Geode (und andere Prozessoren)

Mit einigen wenigen Compileroptions läßt sich die durch SvxLink hervorgerufene CPU-Last erheblich reduzieren. Ändern Sie hierfür den Parameter RELEASE_CFLAGS in der Dateimakefile.cfg im SvxLink-Source-Verzeichnis wie folgt:

Raspberry Pi (ARM-Architektur):

RELEASE_CFLAGS=-g -O2 -mfloat-abi=softfp -mfpu=vfp -mcpu=native

Raspberry Pi (ARM-Architektur) unter Raspbian “wheezy” (Hinweis von Uwe/DC5PI):

RELEASE_CFLAGS=-g -O2 -march=armv6 -mfloat-abi=hard -mfpu=vfp

Futro A240 (Geode):

RELEASE_CFLAGS = -g -O2 -march=geode -mtune=generic

Diese Angeben wurden experimentell ermittelt und erheben keinen Anspruch auf Vollständigkeit!
Weitere Informationen zum gcc und dessen Compileroptionen können Sie hier finden:

ARM-Prozessor: http://gcc.gnu.org/onlinedocs/gcc/ARM-Options.html
Geode (oder andere): http://gcc.gnu.org/onlinedocs/gcc/i386-and-x86_002d64-Options.html

Audioprobleme beim Raspberry Pi

Falls sich beim Rapsberry Pi Audioprobleme zeigen sollten, so kann man den Tip von Wilm (DL4OCH) ausprobieren. Die Ursache liegt vermutlich in Problemen beim USB2.0 und bewirkt dass Audioausgaben gedropped werden (“zittrig” erscheinen).

mögliche Lösung:
Downgrade auf USB1.1
Schritte:
sudo nano /boot/cmdline.txt
dwc_otg.speed=1
Damit wird der USB-Port auf USB 1.1 gezwungen. sudo reboot nicht vergessen.

Einfaches EchoLink-Logbuch

Um automatisch ein kleines und einfaches Logbuch der EchoLink-Verbindungen erstellen zu lassen kann ein kleines Perl-Skript [1] genutzt werden. Dieses Skript liest das SvxLink-Log aus und erstellt daraus eine Logbuchdatei im konfiguriertem Verzeichnis.

Bedingungen:
- auf Ihrem System muß *perl* installiert sein
- Sie müssen SvxLink so betreiben, dass es die Ausgaben in ein Logfile schreibt. Wenn Sie SvxLink auf der Konsole starten werden die Ausgaben NICHT in das SvxLink-Logfile geschrieben

Installation:
1) Laden Sie das Skript herunter und installieren Sie es unter z.B. /usr/bin
2) machen Sie das Skript ausführbar mit *chmod 755 el_logbook.pl*
3) Ändern Sie in der svxlink.conf den Parameter TIMESTAMP_FORMAT="%d.%m.%Y %H:%M:%S", damit die Datums/Zeit-Einträge in deutscher Schreibweise erfolgen
4) im Skript selbst können die Pfade für das svxlink-log und das Logbuch festgelegt werden
5) Wenn Sie eine automatische Erstellung des Logbuchs wünschen, sollten Sie entweder einen Eintrag in der crontab vornehmen oder logrotate nutzen
a) mittels crond:
in */var/spool/cron/tabs/svxlink* einen Eintrag hinzufügen, z.B.:
0 0 * * * /usr/bin/el_logbook.pl 1>/dev/null 2>/dev/null
Damit erfolgt die Erstellung jeden Tag um 0Uhr.

b) mittels logrotate:
Die bessere und sichere Variante ist die mittels *logrotate*. Dann wird immer kurz vor dem Rotieren der SvxLink-Logdatei ein Logbuch erzeugt und somit gehen keine Einträge verloren! Hier ein Beispiel für einen logrotate-Eintrag (/etc/logrotate.d/svxlink):

```
/var/log/svxlink {

    missingok

    notifempty

    daily

    create 0644 svxlink daemon

    prerotate

       /usr/bin/el_logbook.pl

    endscript

    postrotate

        killall -HUP svxlink

    endscript

}
```

Downloads:
[1] EL-Logbuch-Skript (Perl erforderlich)

SvxLink fernsteuern

SvxLink läßt sich per *netcat* wie folgt fernsteuern:
1) SvxLink auf dem Server mit

```
nc -lk 10000|svxlink
```

starten. Die 10000 ist der Port auf dem netcat lauscht bzw. die Kommunikation zwischen SvxLink und dem Steuer-PC abläuft.

2) Vom Steuer-PC aus kann SvxLink dann mit z.B.

```
echo -n "2#1#"|nc svxlink-server-ip 10000
```

ferngesteuert werden wobei *svxlink-server-ip* die IP-Adresse/Hostname des Rechners ist auf dem svxlink läuft.

Startskripte

Die Applikation svxlink sollte nicht als root laufen, das ist absolut nicht notwendig. Legen Sie daher eine Gruppe svxlink und den User svxlink an, führen Sie als root aus:

```
groupadd svxlink

useradd -G svxlink -g svxlink svxlink
```

Vergeben Sie dem Nutzer ein Passwort, bitte nicht so etwas wie abc123!

Die neusten openSuSE-Versionen nutzen den systemd um das Starten und Stoppen der Applikationen zu realisieren. Startskripte in /etc/init.d sind dann nicht mehr erforderlich. Die Einrichtung ist sehr einfach. Erstellen Sie im Verzeichnis eine Date svxlink.service mit folgendem Inhalt:

```
[Unit]
Description=SvxLink Server
After=network.target sound.target
[Service]
Restart=always
KillMode=process
ExecStart=/usr/bin/svxlink --config=/etc/svxlink/svxlink.conf --logfile=/var/log/svxlink
RestartSec=5
TimeoutSec=5
User=svxlink
Group=svxlink
Environment="HOME=/home/svxlink"

[Install]
WantedBy=multi-user.target
```

Aktivieren Sie das Starten des SvxLink-Servers mit:

```
systemctl enable svxlink.service
```

und starten Sie svxLink mit:

```
systemctl start svxlink.service
```

Der systemd überwacht die Funktion dieses Dienstes und startet ihn im Bedarfsfall neu.

Und hier noch ein Skript für Debian, dieses Skript bitte in *svxlink* umbenennen und nach*/etc/init.d/* kopieren:
Download SvxLink-Startskript für Debian
Dieses Skript muß ausführbar sein:

```
chmod 755 /etc/init.d/svxlink
```

und dann noch aktiviert werden, damit es beim Systemstart automatisch ausgeführt wird:

```
sudo update-rc.d svxlink defaults
```

Wieterführende Informationen zu Diensten kann man hier findenhttp://wiki.ubuntuusers.de/Dienste

Links

Stationen / Hams, die SvxLink betreiben:

- SM0SVX - The big cheese 😉 http://www.ludd.luth.se/~blomman/
- DB0WA http://www.db0wa.de
- DB0DIN http://www.db0din.de/
- DB0PER http://www.db0per.de/
- DB0BLO http://www.ov-d20.de/smb.htm
- DB0HHW http://db0hhw.blogspot.com/
- DB0TUD http://www.qsl.net/dh1dm/db0tud/
- DB0HRC http://www.darc.de/distrikte/w/34/db0hrc-bad-duerrenberg/
- DB0WSF http://www.darc.de/distrikte/w/34/relaisstation-db0wsf/
- DB0DBT http://www.darc.de/distrikte/w/34/db0dbt-allstedtflugplatz/
- DB0HFT http://www.db0hft.de/
- DL1BIR http://www.mydarc.de/dl1bir/
- DB0HDF http://www.db0hdf.de/
- DB0HSN http://df0fn.hsnr.de/db0hsn/db0hsn.htm
- DM0LEI http://wiki.dh8tom.net/doku.php?id=dm0lei
- DB0CSD http://www.dm2kl.de/relais.html
- DB0ZD http://www.db0zd.de.vu/
- DB0LHR http://www.funkamateure-lahr.de/htm/db0lhr.htm
- DB0MYK http://prgm.org/db0myk
- DB0FBG http://www.qsl.net/dl0fg/db0fbg.html
- DL0RI http://www.avsk.net/165.html
- DB0PB http://db0ug.ulrich-klute.de/index.php?option=com_co…
- DB0BUL http://www.dl8rds.de/index.php/The_Echolink_Node_DB0BUL
- DB0TIT http://www.ov-a34.de/index.php/echolink
- DB0KOE http://www.db0koe.de/cms17/…/16-23cm-demnaechst-mit-svxlink
- DO3DT http://home.arcor.de/db0dxm/index.html
- OK1PJV http://www.cbradio.cz/echolink/ok1pjv-l.php3
- OE5XYP http://www.adl509.at/…/6-meter-umsetzer/oe5xyp/
- OE5XOL http://wiki.oevsv.at/index.php/70cm_Relais_OE5XOL
- PE1JLX http://www.splint.nl/
- SK3GW http://sk3w.se/sk3gw-repeater
- SL0ZS http://www.stockholm.fro.se/paketradio/sl0zs/index.html
- S50EDX http://www.s50e.si/?page_id=71
- S55VTO http://rpt.hamradio.si/?modul=galerije&site=0&id=0-39-
- TA3CP http://ta3cp.blogspot.de/search/label/svxlink
- F1SMF http://f1smf.free.fr/articles.php?lng=fr&pg=20
- ON0LN http://www.on8ar.be/index.php?option=com_content&amp…
- WB0RXX http://www.hacem.org/svxlink.html
- 4Z7GAI http://guysoft.wordpress.com/2011/08/08/svxlink/
- KJ6ZD-L http://www.svxlink.net

Weiterführende Informationen:

- SvxLink.de Webbook Stand 08/2013
- http://www.dl8rds.de/index.php/A_SVXLink_Server_based_on_the_BeagleBoard
- Die offizielle SvxLink Seite bei Sourceforge (Engl.)
- Vortrag von Prof.-Dr. Michael Hartje zum Thema “Digitale Amateurfunkrelais” auf der HAMRADIO 2012
- Youtube Video, Prof. Dr. Michael Hartje / DK5HH, HAMRADIO 2012
- Linux-Livesystem mit SvxLink – Vollständiges Linuxsystem mit graphischer Oberfläche und bereits enthaltenem SvxLink
- Echolink bei Wikipedia http://de.wikipedia.org/wiki/Echolink

HowTo

Modul EchoLink:

Wie kann ich bei einer eingehenden EchoLink-Verbindung einen Rufton aussenden um beispielsweise ein Relais zu öffnen?

Erstellen Sie eine tcl-Datei (z.B. el-beep.tcl) nach dem lokale-Prinzip mit folgendem Inhalt:

```
namespace eval EchoLink {

  proc remote_connected {call} {

    playTone 1750 100 1000;

    playMsg "connected";

    spellEchoLinkCallsign $call;

    playSilence 500;

  }

}
```

Starten Sie SvxLink neu.

Repeater-/Simplexbetrieb:

Wie kann die Tonhöhe/-länge des Roger-Pieps geändert werden?

Erstellen Sie eine tcl-Datei (z.B. rogerbeep.tcl) nach dem lokale-Prinzip mit folgendem Inhalt:

```
namespace eval Logic {

  proc send_rgr_sound {} {

    variable sql_rx_id;



    playTone XXXX 500 YYYY; # hier XXXX durch die neue

                            # Tonhöhe ersetzen (Bereich: 300-3000)

                            # und YYYY durch die neue Tonlänge in ms

    playSilence 200;



    for {set i 0} {$i < $sql_rx_id} {incr i 1} {

        playTone 880 500 50;

        playSilence 50;

    }

    playSilence 100;

  }
```

```
}
```

Starten Sie SvxLink neu.

Wie kann man periodische Aussendungen (Zeit,Call,…) auf einem Link unterbinden während eine EL-Station bzw. eine lokale Station sprechen?
Erstellen Sie eine tcl-Datei (z.B. stop_announce.tcl) nach dem lokale-Prinzip mit folgendem Inhalt:

```
namespace eval EchoLink {

  variable el_rx 0;

  proc is_receiving {isrec} {

    set el_rx $isrec;

  }

}

namespace eval Logic {

  variable rx_state 0;

  proc send_long_ident {hour minute} {

    if {$EchoLink::el_rx != 0 || $rx_state != 0} {

      exit;

    }

    # kopieren sie den Inhalt der Prozedur send_long_ident der

    # Original-Logic.tcl bzw. Ihre Änderungen hier herein

    #

  }



  proc squelch_open {rx_id is_open} {

    variable sql_rx_id;

    set sql_rx_id $rx_id;

    set rx_state $is_open;

  }

}
```

Starten Sie SvxLink neu. Achten Sie darauf, dass Sie wahrscheinlich diese Prozeduren schon vorher an Ihre Bedürfnisse angepaßt haben und Sie diese Änderungen hier “nachziehen” müssen.

Probleme und Hilfe

Es gibt einen Weg um mit der SvxLink-Gemeinde Kontakt aufzunehmen: durch eine Mailingliste. Die Adresse dieser Mailingliste ist svxlink-devel@lists.sourceforge.net.
Was auch immer ihr Anliegen ist, bitte nutzen Sie hierfür die Mailingliste. Nutzen Sie NICHT Echolink und senden Sie auch keine direkte Mail an den Verfasser. Diese Mailingliste ist nicht stark frequentiert, so dass Sie keine Bedenken haben müssen sich dort zu registrieren.

Es gibt eine Reihe von Gründen weshalb nur die Mailingliste zur Unterstützung bei Fragen und Problemen genutzt werden sollten:

Es gibt mehr Leute, die helfen könnten. Ich selbst (Tobias, SM0SVX, d.Red.) nutze nur Fedora Linux aber es gibt hier Leute die haben SvxLink unter verschiedenen Distributionen laufen

Dokumentation. Alle eMails, die an die Liste gesendet werden, werden im Malinglisten-Archiv gespeichert. Eine gute Empfehlung zur Suche nach Lösungen ist, zunächst im Mailinglistenarchiv nachzuschauen.

Andere Leute die dieser Liste beitreten, können sich an der Diskussion beteiligen

Es gibt noch eine andere Mailingliste svxlink-announce@lists.sourceforge.net welche noch weniger frquentiert ist. Bei jeder neuen Version wird eine einzelne eMail versendet. Das ist wirklich nicht viel. Wenn Sie bereits der svxlink-devel-Liste beigetreten sind, dann gibt es keine Notwendigkeit auch der svxlink-announce-liste beizutreten.

Um den Listen beizutreten klicken Sie bitte hier. Beachten Sie bitte, dass in den Mailinglisten Englisch gesprochen wird!

Fehler melden

Fehler sollten dem SourceForge Fehler-Tracking-System gemeldet werden. Es kann manchmal schwierig sein, einen gute Fehlermeldung zu schreiben aber ich versichere Ihnen das es noch schwieriger ist eine schlechte Fehlerbeschreibung zu verstehen. Wenn Sie also einen Fehler finden, nehmen Sie sich bitte die Zeit die folgenden Schritte abzuarbeiten:

1) Schauen Sie im Fehler-Tracker nach um zu sehen, ob dieser Fehler schon einmal gemeldet wurde

2) Lesen Sie sich die Dokumentation noch einmal durch um auch ganz sicher zu sein, dass Sie nicht eventuell etwas mißverstanden haben. Das schließt die Installationsanleitung, die SvxLink Server-Dokumentation und die Qtel-Dokumentation, abhänging davon in welcher Applikation Sie den Fehler gefunden haben, mit ein.

3) Versuchen Sie wenn möglich den Fehler zu reproduzieren. Die Art und Weise wie ein Fehler zu reproduzieren ist macht es dem Entwickler einfacher den Fehler zu finden und zu beseitigen.
Wenn es nicht möglich ist den Fehler zu reproduzieren, versuchen Sie sich daran zu erinnern, welche genauen Handlungen dem Fehler vorausgingen. Fügen Sie der Meldung eine genaue Schritt-für-Schritt-Beschreibung hinzu damit man den Fehler eventuell reproduzieren kann.

4) Schreiben Sie einen Fehlerbreicht im Fehler-Tracker. Der Fehlerbreicht sollte folgendes beinhalten:

Eine detailierte Beschreibung des Fehlers. Schreiben sie nicht einfach “Das funktioniert nicht”. Erläutern sie auf unter welchen Umständen es nicht funktioniert. Es kann sehr viele Möglichkeiten geben, weshalb eine bestimmte Funktion nicht funktioniert.

Wenn möglich, schreiben Sie wie der Fehler reproduziert werden kann.

Wenn Sie diese einfachen Regeln befolgen, können Fehler viel schneller gefunden und behoben werden.